STOCHASTIC COMPUTATION FOR ENERGY-EFFICIENT ROBUST ULTRA-LOW-POWER PLATFORMS

BY

RAMI A. ABDALLAH

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Electrical and Computer Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2012

Urbana, Illinois

Doctoral Committee:

Professor Naresh R. Shanbhag, Chair
Professor Douglas L. Jones
Professor Philip T. Krein
Assistant Professor Rakesh Kumar

ABSTRACT

Next-generation ubiquitous computing promises new levels in immersion and seamless technology integration enabled through a profusion of embedded signal processing (DSP)-heavy ultra-low-power (ULP) platforms. This dissertation proposes an holistic integrated stochastic computing approach to enable the design of next-generation ULP platforms that operate dramatically closer to the limits of the achievable robustness-energy-performance envelope over a highly unreliable device fabric.

Stochastic computing was shown to be an elegant design approach for energy-efficient and robust systems-on-a-chip (SoC) in superthreshold applications. This dissertation studies and extends the application of stochastic computing to the minimum-energy operating point (MEOP), which is known to occur in the subthreshold regime. Analysis, architecture and circuit-level simulations, and integrated circuit (IC) measurements in a 45-nm CMOS technology, are employed to study the stochastic-subthreshold design space. Energy savings of 28% to 54% beyond minimum achievable energy E_{min} at the conventional (error-free) MEOP, along with 380× to 850× increase in pre-correction error rate (p_η) handling capability, are demonstrated in the presence of voltage and process variations. A stochastic computing-based biomedical processing IC is designed at the MEOP. The supply voltage of the prototype IC can be scaled to 15% below its critical (error-free) value of 0.4 V, while compensating for a $p_\eta = 58\%$, improving the heart-beat detection accuracy by 19×, and achieving 28% E_{min}-energy savings over conventional MEOP processors. This IC consumes 14.5 fJ/cycle/1k-gate and exhibits 4.7× better energy efficiency than the state-of-the-art while tolerating 16× more voltage variations.

To further enhance system-energy efficiency, this dissertation proposes an integrated design approach for ULP platforms by jointly optimizing the design of the compute cores and the energy-delivery subsystem. Joint core architecture and DC-DC converter design techniques are proposed in order to minimize the total system energy consumption. Results show a 45.5% system-energy savings and a 2.3× improvement in the efficiency of energy-delivery subsystem over the conventional case where the system is operated at the core MEOP while ignoring DC-DC converter losses.

This dissertation makes a contribution, to the portfolio of stochastic computing techniques, referred to as *likelihood processing* (LP). LP exploits hardware error statistics to generate reliability information or confidence level on the output bits of a compute block in a statistically optimal manner and with a low complexity. The benefits of LP are demonstrated in the design of a 45-nm discrete-cosine transform (DCT) codec, which can be employed as a hardware accelerator in a ULP platform. LP is shown to tolerate 5× to 100× greater vales of p_η, and achieve 15% to 71% energy savings, when compared with existing techniques.

Stochastic computing advocates an explicit characterization and processing of error statistics at the architectural/system level. To support this need, this dissertation proposes a unified framework with a generalized statistical error characterization methodology. The proposed framework and methodology are analyzed and verified for a number of 45-nm DSP kernels. Furthermore, design diversity techniques are introduced in order to engineer favorable spatially-independent error-statistics and aid the robustness and implementation of stochastic computing.

The proposed design principles and demonstrated energy and robustness benefits in this dissertation can be generalized beyond ULP platforms to modern high-throughput computational platforms. This is timely, because such platforms are moving the direction of becoming a heterogenous many-core SoC. Thus, the work in this dissertation, and stochastic computing in general, can provide stochastic accelerator cores for integration with conventional cores on to such a platform.

To my mother, my father, and my advisor

ACKNOWLEDGMENTS

And I say that life is indeed darkness save when there is urge,
And all urge is blind save when there is knowledge,
And all knowledge is vain save when there is work,
And all work is empty save when there is love;
And when you work with love you bind yourself to yourself, and to one another, and to God.
Work is love made visible... And if you sing though as angels, and love not the singing, you muffle man's ears to the voices of the day and the voices of the night.[1]

For this quote, I would like to thank all those who have inspired me, challenged me, taught me, or helped me love my work.

Foremost, my deepest gratitude goes to God, my parents, and my advisor. My parents have never hesitated to sacrifice anything for the well-being of their children. Their constant love and encouragement have always pushed me to do my very best. My advisor, Professor Naresh Shanbhag, has not been only a teacher and a mentor throughout these years but also a father. I am confident that his role in my life will go much beyond these few years. I am greatly indebted to him for all the advice, knowledge, responsibilities, and opportunities that he gave me in school and life. The most important things that I learned from him go beyond this dissertation, and I will always cherish them. His attitude towards life, passion to work, creativity, diversified knowledge, high standards, and meticulousness have been and

[1] An excerpt from "*The Prophet*" by Gibran Khalil Gibran.

will always be a source of inspiration and a lifelong example. I am fortunate to have him and my parents, and to them I dedicate this dissertation hoping that they find it worthy of their expectations of me.

I would also like to thank Professors Douglas Jones, Rakesh Kumar, and Philip Krein for the many insightful discussions and input during weekly group meetings and agreeing to be on my thesis committee. Their comments and suggestions have greatly helped improve this thesis. My sincere thanks also to Yu-Hung Lee for his help on error statistics simulations in Chapter 6, my research group colleagues for their feedback and input during group meetings, and the circuits group staff, Carolyn Genzel and Loren Heal, for their help and support. I am also grateful to my sister and brothers for being there for me, and my labmates and friends at Illinois who made me feel at home while being away from home. In particular, I would like to mention Dr. Ali Bazzi, Dr. Marc Ghossoub, and Dr. Rajan Narasimha.

The support of the Gigascale Systems Research Center (GSRC), one of six research centers funded under the Focus Center Research Program (FCRP), a Semiconductor Research Corporation (SRC) entity, and the support from Texas Instruments, Inc., are gratefully acknowledged.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ANT	algorithmic noise-tolerance
BER	bit error rate
BPP	bit probability profile
BTWC	better than worst case
C-MEOP	core minimum energy operating point
CAD	computer-aided design
CMOS	complementary MOSFET oxide semiconductor
CBA	carry-bypass adder
CPU	central processing unit
CSA	carry-select adder
CVD	cardiovascular disease
DCT	discrete cosine transform
DF-FIR	direct-form finite input response
DMR	dual-modular redundancy
DSP	digital signal processing
DVS	dynamic voltage scaling
ECG	electrocardiograph
FIR	finite input response
FOS	frequency overscaling
IC	integrated circuit

IDCT	inverse discrete cosine transform
KL	Kullback-Leibler
LP	likelihood processing
MEOP	minimum energy operating point
MSE	mean square error
NMR	N-modular redundancy
PMF	probability mass function
PE	processing element
PoFF	point of first failure
PSNR	peak signal-to-noise ratio
PTA	Pan-Tompkins algorithm
PVT	process, voltage, and temperature
RC	reconfigurable core
RCA	ripple carry adder
RTL	register transfer language
PC-MEOP	parallelized core minimum energy operating point
S-MEOP	system minimum energy operating point
SS-MEOP	stochastic system minimum energy operating point
SC	single core
SNR	signal-to-noise ratio
SoC	system-on-chip
SOI	silicon-on-insulator
SSNOC	stochastic sensor network-on-chip
TF-FIR	transposed-form finite input response
TMR	triple-modular redundancy
ULP	ultra-low-power
VOS	voltage overscaling

CHAPTER 1

INTRODUCTION

As we move into a world of immersive and ubiquitous computing, we will witness a multitude of embedded processors that are going to transform the way we interact with the physical world. Sensing, surveillance, and media-rich immersive computing will constitute a large part of next-generation applications in environmental monitoring, supply chain management, traffic control, power distribution, smart automotives and avionics, health care , entertainment, and defense applications. Such applications will be characterized by increased functionality being embedded in multiple tiny ultra-low-power (ULP) devices/nodes giving rise to a *sensory swarm* (see Fig. 1.1(a)) [1], which will act as virtual eyes, ears, and hands for data collection and analysis. For example, Fig. 1.1(b) shows an example of body-area sensor network (BSN) [2] which consists of multiple embedded sensing and processing nodes on, near, or within human body, BSN promises new levels of immersion and novel uses in health-care, medical diagnostic services, and entertainment. It is predicted that the number of embedded processors per person will exceed 1000 by 2015 [3]. Form-factor and energy are the main design drivers in these applications for seamless integration and extended lifetime since frequent battery replacement with thousands of embedded processors is not a feasible option, or these processors are being operated on scavenged energy sources [4].

A growing concern with increased integration and complexity in these applications is reliability, since increased process, voltage and temperature (PVT) variations, leakage, soft errors, and noise in sub-45-nm process technologies [5] are conspiring to offset the energy and area benefits of feature-size scaling. Current design philosophy addresses the reliability problem at the expense of power or energy consumption by targeting worst-case variations

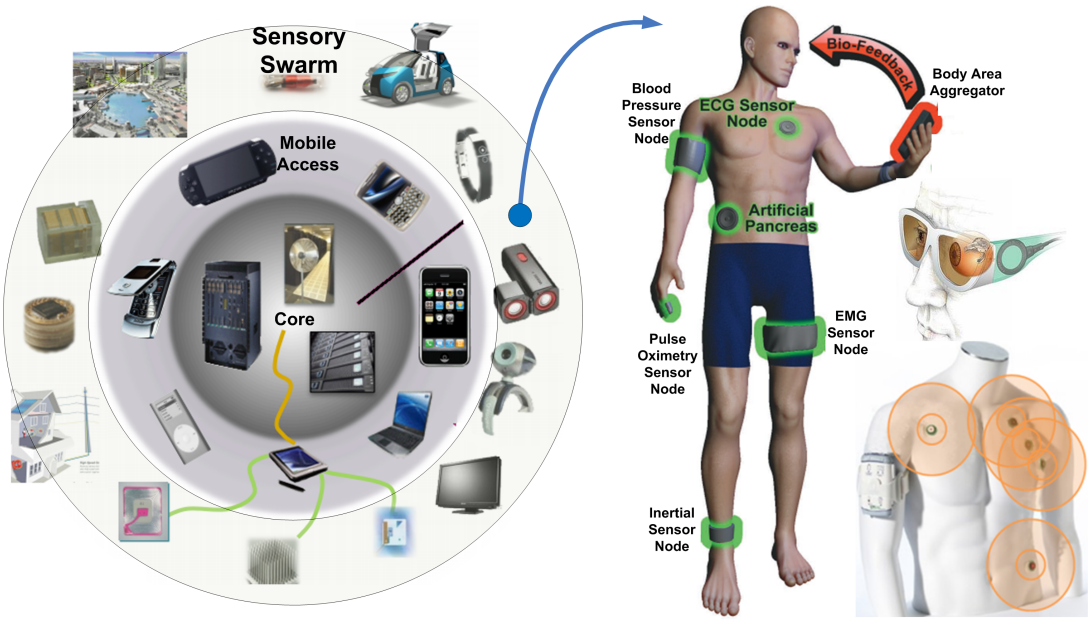


Figure 1.1: Ubiquitous computing: (a) infrastructure [1] and (b) an example: body-area sensor network.

and scenarios. In fact, the International Technology Roadmap for Semiconductors (ITRS) [6] has stated the achievement of reliability and energy-efficiency as two of the Grand Challenges facing the semiconductor industry. ULP platforms are typically operated in subthreshold (below the threshold voltage (V_{th}) which is typically in the few hundred mV's range) to save energy. This leads to increased PVT variations due to the exponential relation between current and supply voltage in subthreshold. For example, in 65-nm CMOS, worst-case 3-σ gate delay is around 3 orders-of-magnitude from the nominal case (see Fig. 1.2) [7].

Further complicating the problem of energy in ULP platforms is the low efficiency and increased size of the energy-delivery subsystem (DC-DC converter). Technology scaling and energy-aware computing techniques have made it possible to operate cores at low V_{dd}. Low V_{dd} operation, however, exacerbates the energy-delivery network and leads to the deployment of multiple large programmable voltage regulators/DC-DC converters in modern platforms (see Fig. 1.3(a)). In a typical sensor node platform (see Fig. 1.3(b)), depending on the throughput requirement, a single or multiple programable DC-DC converters adjust

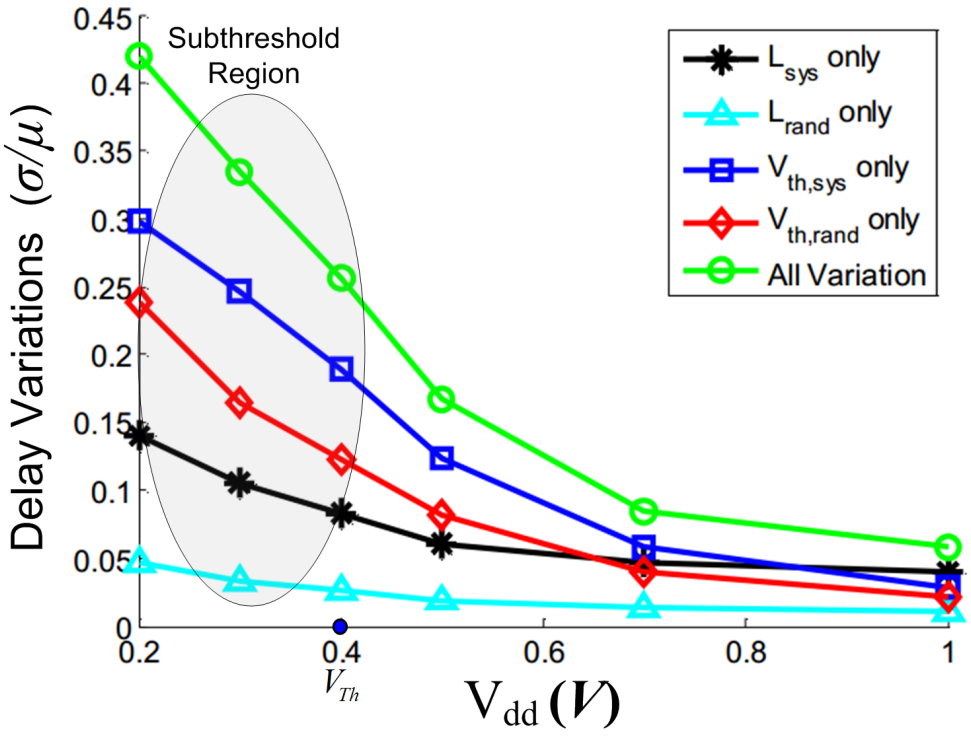


Figure 1.2: The 1-σ gate delay variations due to random and systematic process variations in 65-nm CMOS [7].

the core supply voltage V_{dd} from an external battery source (V_{bat}) depending on the required throughput. For example in subthreshold operation, the DC-DC converter is required to convert a battery voltage $V_{BAT} = 1.2\,\mathrm{V}$ to a core voltage $V_{dd} < 0.5\,\mathrm{V}$. This severely reduces the converter energy efficiency η_{DC}, sometimes below 40% as shown by integrated circuits (IC) measurements in Fig. 1.3(c) [8] and reported in [9]. Addressing the energy-delivery system in ULP sensory application is made even more challenging by the small form factor requirements and the extremely large variations in workload characteristics (up to four orders of magnitude [10]) due to the event-driven nature of sensory ULP applications. A ULP platform architecture (see Fig. 1.3(b)) consists mainly of 1) a set of sensors with signal-processing kernels to detect an interrupt, 2) an event or general-purpose processor to manage interrupts, perform complex detection, and activate appropriate modules, 3) a set of hardware accelerators for enhanced performance and energy savings, 4) a memory subsystem, and 5) a communication kernel/radio, in addition to the DC-DC converter.

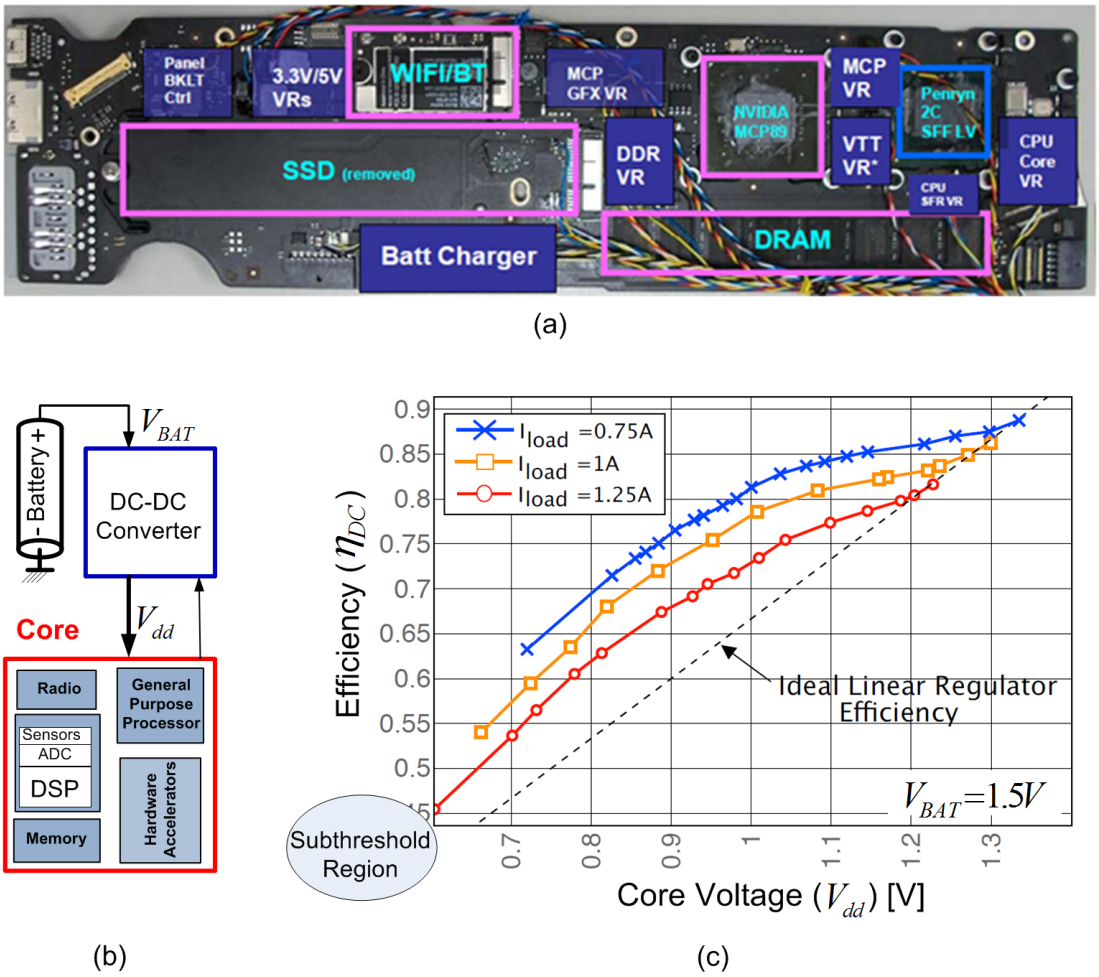


Figure 1.3: Energy delivery in ULP platforms: (a) power delivery network in a *MAC-Book Air* [11] (DC-DC converters/voltage regulators (VRs) are shown in blue), (b) sensor node block diagram, and (c) measured DC-DC convertor efficiency in 45-nm CMOS [8].

These subsystems have different duty cycles and performance requirements depending on the environmental settings and workload. For example, the sensors and signal processing modules are activated all the time and operate in subthreshold with very low frequencies to match the observed phenomenon while consuming negligible power. At the other extreme, the hardware-accelerators are power-gated most of the time and activated depending on the complexity and severity of the required processing.

Thus, the design of robust and energy-efficient ULP platforms is an important area of research and this dissertation presents novel solutions by simultaneously addressing energy

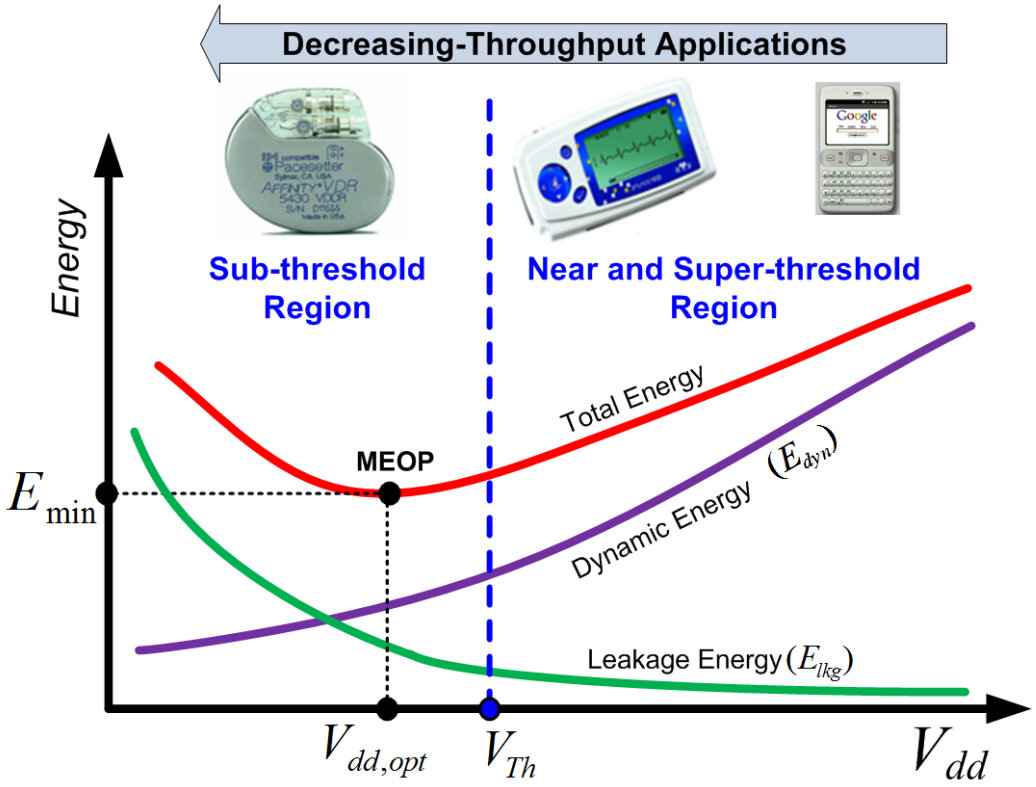


Figure 1.4: Energy under dynamic and aggressive voltage scaling.

efficiency and reliability in ULP platforms while taking into consideration the energy-delivery subsystem.

1.1 Past Relevant Work

This section describes the past work addressing the three problems: reliability, energy efficiency, and energy-delivery. In the past, these three problems have been addressed independently leading to sub-optimal, and sometimes, contradictory design principles.

1.1.1 Low-Power Design

Low-power digital circuits and system design is a mature topic of research since the early 1990's [12–15]. Supply voltage scaling [12], body biasing [16], transistor sizing [17], clock-gating [18], and reconfiguration [14] [19] are commonly employed energy reduction techniques

in practice today. Voltage scaling has been the workhorse technique for energy-efficient operation. Dynamic voltage scaling (DVS) [20–22] is employed to match the voltage and frequency of operation to the application requirements in order to minimize energy consumption in the superthreshold regime.

In the subthreshold regime, aggressive voltage scaling (see Fig. 1.4) where the core supply voltage $V_{dd} < V_{th}$ is a well-established design approach for low-throughput (hundreds of Hz to few MHz) applications. Reducing the supply voltage V_{dd} results in quadratic reduction in *dynamic* energy consumption E_{dyn}. However, as V_{dd} is reduced below V_{th}, delay, and hence the leakage energy E_{lkg}, increases exponentially and quickly becomes comparable to E_{dyn}. This trade-off between E_{dyn} and E_{lkg} is well-studied [23–26], and results in a minimum energy operating point (MEOP). The MEOP is characterized by the tuple $(V_{dd,opt}, f_{opt}, E_{min})$, where $V_{dd,opt}$ and f_{opt} are the energy-optimum supply voltage and frequency of operation, respectively, resulting in the minimum achievable system energy E_{min}. In the last few years, several ICs [27–29] and embedded processors [30–32] operating at the MEOP have been demonstrated, and biomedical sensing and processing platforms [33–38] that exploit dynamic or aggressive voltage scaling have appeared. Several circuit-level techniques [39], such as multi-threshold CMOS [40], sleep transistor [41], and body biasing [42], and device-level techniques such as channel and gate engineering [39] have been proposed to reduce subthreshold leakage and total system energy. In addition, architectural techniques such as pipelining [28] have been employed to reduce E_{min}.

These low-power design techniques address energy reduction at the expense of reducing performance (speed). These also lead to greater sensitivity to PVT variations due to the use of smaller devices or smaller logic depth. Thus, the conventional worst-case design philosophy adopted currently tends to offset the energy benefits promised by such techniques. Furthermore, these techniques address energy efficiency of the core only, and ignore the impact of low V_{dd} and low load current on the efficiency of the energy-delivery subsystem.

1.1.2 Robust System Design

Design of reliable systems dates back to Von Neumann [43] in 1950, where logic networks composed of noisy/probabilistic gates were considered and signal replication with majority voting was proposed to increase resiliency. A number of logic-level techniques for robust design have been proposed since then. In [44], Markov random networks are employed to design robust logic network. This implementation, though quite robust to input voltage noise, has a large overhead in terms of transistor count. In [45], *stochastic logic* is proposed whereby Von Neumann's N-wire bundle representation of Boolean variables is employed. It is assumed that the logic is error-free, i.e., deterministic logic (error-free hardware) operating on stochastic signals. These techniques provide error resiliency, at the expense of a large gate-count overhead ($> 5\times$) which limits their energy efficiency.

At architecture-level, N-modular redundancy (NMR) [46] is a commonly employed fault-tolerant technique where N-way replication of processing elements is followed by majority vote. NMR leads to an N-fold complexity and power overhead. Similarly, techniques such as checkpointing [47], and coding techniques [48] [49] have been proposed, each of which, though effective in enhancing robustness, incurs a significant energy cost.

The problem of increased circuit sensitivity to PVT variations in subthreshold has also been addressed at circuit and architectural levels. Circuit-level techniques, such as body-biasing [50], non-ratioed logic styles [51], and transistor sizing [26], reduces V_{th} variations and consequently delay variations. Architectural techniques such as increased pipelining depth [52] have been applied in subthreshold to mitigate the effect of PVT variations across longer critical oath delays. However, these circuit and architectural-level techniques either significantly increase energy or lower the performance.

The robustness overhead in each of the approaches described above aggravates the energy-efficiency problem. The main challenge in robust system designs is to achieve robustness with a low error-compensation overhead. Recently, deterministic microarchitecture error

compensation techniques [53–55], which employ local error detection and global error correction via architectural replay, have been proposed. Deterministic microarchitecture-level error correction [53–55] is able to correct for error rates (percentage of clock-cycles in which the output is in error) $p_\eta < 0.1\%$ while achieving an energy efficiency of less than 15% over point-of-first-failure (PoFF). For example, RAZOR-II [54] operates at an error rate of $p_\eta = 4 \times 10^{-4}$ while achieving an energy savings of 5% beyond PoFF, and error detection sequential (EDS) and tunable replica circuits (TRC) [55] achieve 7% energy reduction over PoFF while tolerating a pre-correction error rate $p_\eta < 0.001$. Deterministic error compensation techniques rely on correcting for worst-case PVT variations and achieving 100% correctness which may not be required at the application level. This severely limits their design margins and energy efficiency. They miss the opportunity presented by a large class of next generation applications where a relaxed definition of "correctness" is adopted, and thus admitting an expanded set of acceptable outputs.

1.1.3 Design of Integrated Energy-Delivery Subsystems

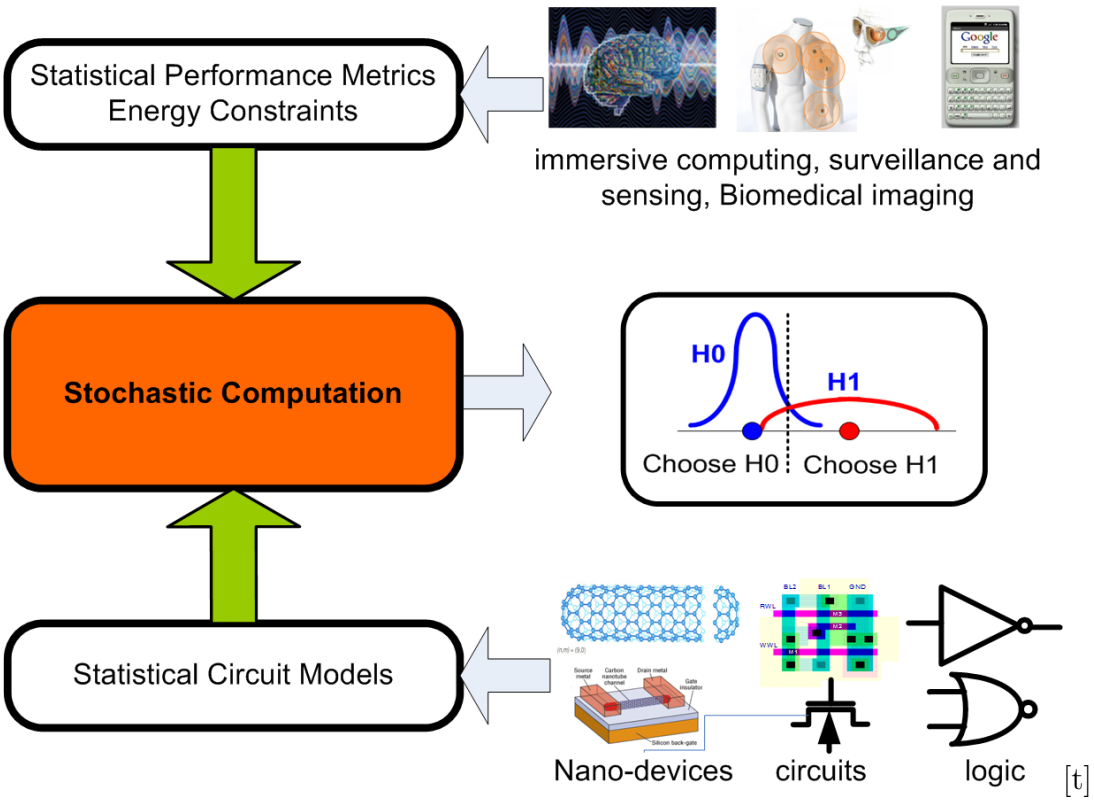
The design of high-efficiency DC-DC converters has been extensively explored [56–59] with converter efficiencies $\eta_{DC} > 0.9$ being achieved for constant load currents, and for high-power superthreshold core operation. However, as we have illustrated, the efficiency η_{DC} varies significantly with variations in the load current, and can fall below 40% [8] [9]. Converters for subthreshold applications have appeared recently in the literature [60]. These tend to employ switched capacitor topology to enable full on-chip integration. However, switched-cap converters suffer from either poor load regulation when V_{dd} is adjusted, or significant losses as the load varies. Design of integrated circuits where the DC-DC converter tracks the MEOP of the core as a function of switching activity, has also been done [61]. However, energy-delivery losses are not accounted for. The increased energy loss in the subthreshold region due to the converter was illustrated recently [62]. Furthermore, the CAD community

has been playing a leading role in incorporating models for converter losses [63] [64] and converter transition delay overhead [65], in determining optimal power management policies, and designing power distribution networks for superthreshold operation.

As a result, past work in the design of integrated converters has focused on either design of high-efficiency converters at fixed high-current loads for superthreshold cores, or on the design of converters to track the core $V_{dd,opt}$ in the subthreshold region but ignoring energy-delivery losses. Thus, there is a unique opportunity for the joint design of DC-DC converters and subthreshold cores to minimize total system energy for variable load conditions. This is particularly important in ULP applications where at least up to four orders of magnitude variation is observed [10].

1.2 Stochastic Computing Techniques

A large class of the next-generation applications can be categorized into recognition, mining, and synthesis, where massive data volume needs to be processed [1]. Such applications rely heavily on digital signal processing (DSP) algorithms and employ statistical performance metrics, such as signal-to-noise ratio (SNR), bit error-rate (BER), probability of detection, and many others. The statistical nature of these metrics allows a somewhat relaxed definition of "correctness" where the output may be corrupted by errors. Stochastic computing [66] relies on exploiting such a relaxed definition of "correctness" afforded by emerging applications. Stochastic computing (see Fig. 1.5) matches the statistical attributes of the underlying circuit/device fabric to the statistical nature of the application-level performance metrics. This is accomplished by employing statistical error compensation (SEC) (see Fig. 1.6(a)). SEC exploits the statistical knowledge of the circuit/device fabric error, the input, and the intermediate data processed by the main block in order to correct for the output error approximately at low overhead such that an application performance metric, such as SNR, is maintained within an application dependent tolerance limit $\triangle SNR$. The non-uniformity of



[t]

Figure 1.5: Stochastic computing in ULP platforms: matching noisy circuits to application metric.

the error statistics (see Fig. 1.6(b)) implies that certain error magnitudes are more likely to occur than others. The statistics of the input and the intermediate data enable SEC with low overhead. Several error-resilient techniques have been developed, also based on the concept of stochastic computing such as those in [67–69]. Naturally, there will always be a small class of critical applications such as those in finance/banking, flight control systems, and others where a precise definition of correctness is mandatory, and where stochastic computing may not be applicable in its current form.

Typically, voltage overscaling (VOS) [70] is employed in the main block to reduce its energy. In VOS, the supply voltage is scaled below the critical voltage $V_{dd-crit}$ needed for error-free operation with a fixed operating frequency. When the supply voltage is lower than $V_{dd-crit}$, the circuit will operate slower than the designed margins, and thus timing violations will occur. SEC is then introduced in order to compensate for timing violations

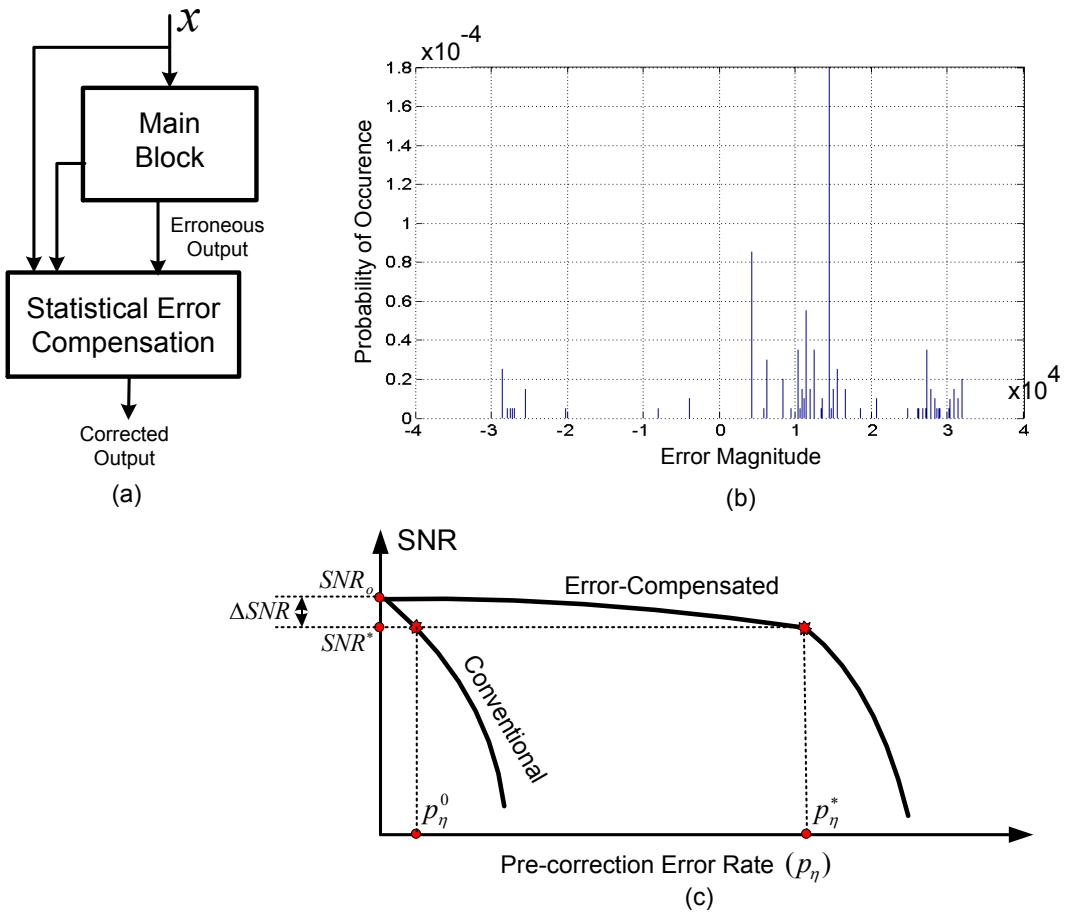


Figure 1.6: Stochastic computing: (a) block diagram, (b) measured error statistics at the output of an FIR filter in a 45-nm CMOS process with V_{dd} scaled 15% below its critical value, and (c) signal-to-noise ratio (SNR) performance at different pre-correction error rates p_η.

(see Fig. 1.6(c)) such that: $SNR^* \approx SNR_o$ and $p_\eta^* \gg p_\eta^o$, $SNR*$ is the SNR of the error compensated system at an error rate p_η^*, and SNR_o is the SNR of the conventional error-free system at an error-rate $p_\eta \approx p_\eta^o$.

1.2.1 Algorithmic Noise-Tolerance (ANT)

Stochastic computing was first proposed in the form of *algorithmic noise-tolerance* (ANT) [70] (see Fig. 1.7(b)). ANT incorporates a *main* block and an *estimator*. The main block is per-

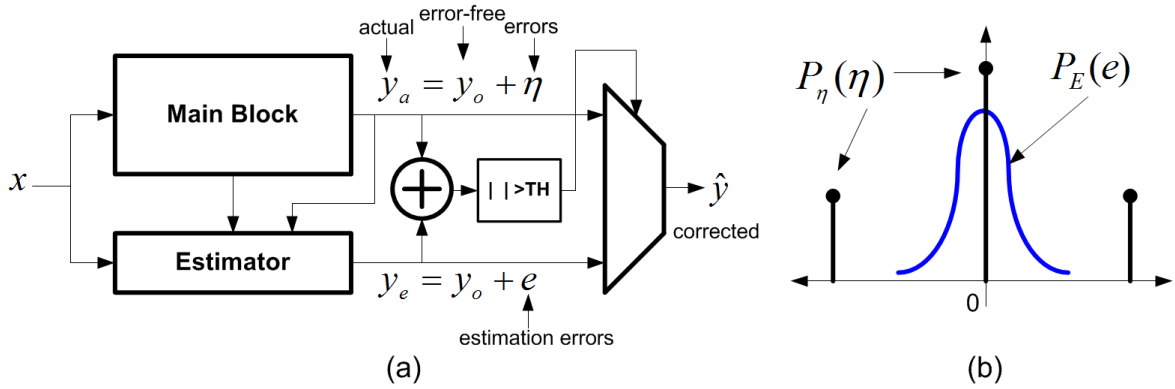


Figure 1.7: Algorithmic noise-tolerance (ANT): (a) framework and (b) error distributions.

mitted to make errors, but not the estimator. The estimator is a low-complexity block (typically 5% to 20% of the main block complexity) generating a statistical estimate of the correct main block output, i.e.,

$$y_a = y_o + \eta \tag{1.1}$$

$$y_e = y_o + e \tag{1.2}$$

where y_a is the actual main block output, y_o is the error-free main block output, η is the hardware error, y_e is the estimator output, and e is the estimation error. Note: the estimator exhibits an estimation error e because it is an approximate version of the main block. ANT exploits the difference in the statistics of η and e (see Fig. 1.7(b)). To enhance robustness, it is necessary that when $\eta \neq 0$, that η be large compared to e. In addition, the probability of the event $\eta \neq 0$, must be small. The final/corrected output of an ANT system $\hat{y}$ is obtained via the following decision rule:

$$\hat{y} = \begin{cases} y_a, & \text{if } |y_a - y_e| < \tau \\ y_e, & \text{otherwise} \end{cases} \tag{1.3}$$

where τ is an application-dependent parameter chosen to maximize the performance of ANT. Under the conditions outlined above, it is possible to show that

$$SNR_{uc} \ll SNR_e \ll SNR_{ANT} \approx SNR_o \tag{1.4}$$

where SNR_{uc}, SNR_e, SNR_{ANT}, and SNR_o are the SNRs of the uncorrected main block (η dominates), the estimator (e dominates), the ANT system, and the error-free main block (ideal), respectively. Thus, ANT detects and corrects errors approximately, but does so in a manner that satisfies an application-level performance specification (SNR). The decision block is designed to be timing error-free at all process corners and voltages because it is a critical block that directly impacts performance (SNR), and typically constitutes less than 5% of the main block complexity. Several low-overhead estimation techniques have been proposed by exploiting data correlation, system architecture, and statistical signal processing techniques [66].

For ANT to also provide energy-efficiency, it is necessary that the errors in the main block be primarily due to enhancement of its energy-efficiency. In practice, these properties are easily satisfied when errors in the main block occur due to VOS or a nominal case design being subjected to a worse case process corner (better than worst-case design (BTWC)). As most computations are least-significant-bit (LSB) first, timing violations due to VOS or BTWC are generally large magnitude most-significant-bit (MSB) errors. Thus, timing violations satisfy the error distribution shown in Fig. 1.7(b).

ANT has been shown to achieve up to 3× energy savings in theory and in practice via prototype IC design [71] for finite impulse response (FIR) filters. ANT has also been employed in the design of error-resilient low-power motion estimators [72] and Viterbi decoders [73] (8000× improvement in BER with 3× improvement in energy savings).

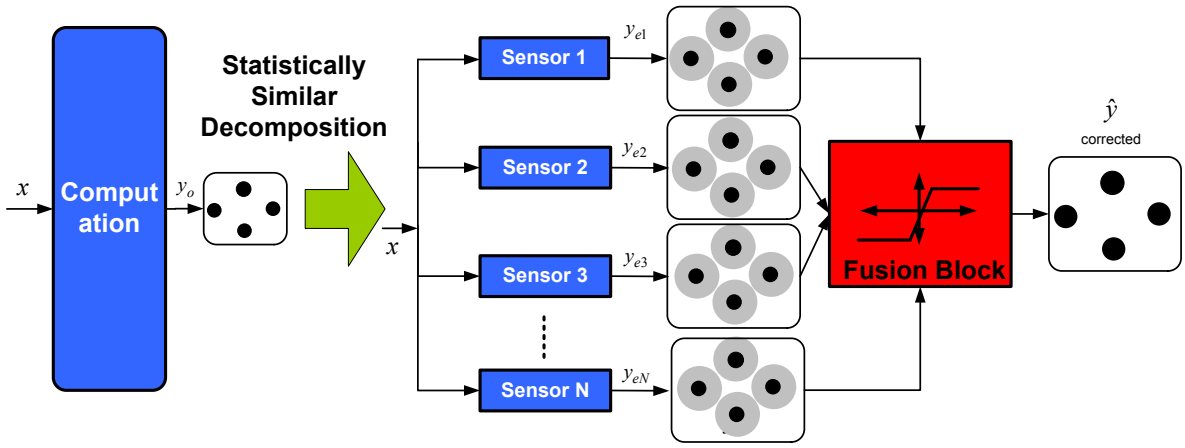


Figure 1.8: The stochastic sensor network-on-a-chip (SSNOC).

1.2.2 Stochastic Sensor Network-on-a-Chip (SSNOC)

SSNOC [74] relies only on multiple estimators or *sensors* to compute, permitting hardware errors to occur (see Fig. 1.8), and then fusing their outputs to generate the final corrected output $\hat{y}$. Thus, the output of the i^{th} sensor is given as

$$y_{ei} = y_o + e_i + \eta_i \tag{1.5}$$

where η_i and e_i are the hardware and estimation errors in the i^{th} estimator, respectively.

If hardware errors are due to timing violations, one can approximate the error term in (1.5) as $(1 - p_\eta)e_i + p_\eta \eta_i$. Such an ϵ-contaminated model lends itself readily to the application of robust statistics [75] for error compensation. SSNOC has been applied to a CDMA PN-code acquisition system [74], where the sensors were obtained through polyphase decomposition of the matched filter. Simulations and IC measurements [76] indicate an 800× improvement in detection probability while achieving up to 40% power savings. A key drawback of SSNOC is the requirement of decomposing computation into appropriate sub-blocks whose output errors have an ϵ-contaminated distribution.

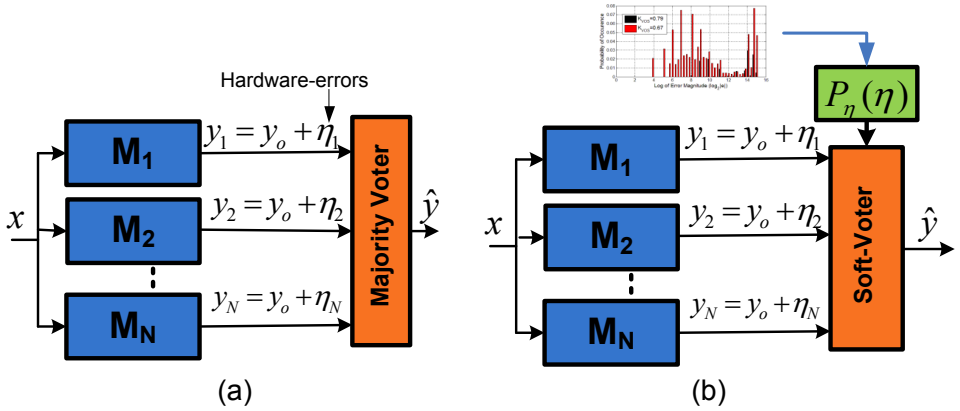


Figure 1.9: Architecture-level error resilient techniques: (a) N-modular redundancy (NMR) and (b) soft NMR.

1.2.3 Soft N-Modular Redundancy (Soft NMR)

ANT and SSNOC rely on certain properties of the distribution of hardware errors η and the estimation error e. For ANT, the distributions of η and e should be sufficiently distinct, and for SSNOC, the composite error distribution should be ϵ-contaminated. More powerful versions of stochastic computation can be developed if error statistics are explicitly employed in computation. In fact, the robustness and energy efficiency of any error-resilient designs depend upon the error statistics, even though error statistics are typically not accounted for in the design. For example, the robustness of NMR depends upon the pre-correction error rate p_η and requires the error events across the redundant modules to be independent [77]. While conventional NMR ignores error statistics, [78] proposed *soft NMR* (see Fig. 1.9) where error statistics, i.e., the probability mass function (PMF) $P_\eta(\eta)$, is explicitly employed to enhance the robustness of NMR.

Structurally, soft NMR differs from NMR in that it incorporates a *soft voter*, which is composed of a *detector*. Soft NMR makes explicit use of two types of statistical information: (1) *data statistics*, and (2) *error statistics*. Data statistics are the distribution of the error-

free output y_o. This is referred to as the *prior* distribution, or prior. Error statistics are the distribution of the errors η_i.

The role of the soft voter in Fig. 1.9(b) is to determine the output $\hat{y}$ that would, on average, optimize a pre-specified performance metric. Detection theory can be employed in order to systematically derive the soft voting algorithm. The detector maps the observations $(y_1, y_2, \ldots, y_N)$ to the "closest" *hypothesis*. Thus, the detection problem requires the definition of a *hypothesis set* $\mathcal{H}$, from which the corrected output $\hat{y}$ is selected. This is done by solving the following problem:

$$\hat{y} = \arg\max_{\mathcal{H}_i \in \mathcal{H}} P(y_1, y_2, ..., y_N | \mathcal{H}_i), \tag{1.6}$$

$$\text{where } \mathcal{H} = \{\mathcal{H}_i\}_{i=1}^{m} \text{ the set of all hypotheses.}$$

As the arg max operation requires a search to be performed over the entire hypotheses space, for practical implementations, the hypothesis space $\mathcal{H}$ needs to be limited. There are several ways to limit $\mathcal{H}$, the simplest being to choose $\mathcal{H} = (y_1, y_2, \ldots, y_N)$.

This dissertation builds and expands on existing stochastic computation applications and techniques in order to design robust and energy efficient ULP platforms while addressing the energy efficiency of both the core and the energy-delivery subsystem.

1.3 Role and Contributions of this Dissertation

Past work in stochastic computing were applied to the superthreshold regime. Furthermore, research in energy efficiency addresses energy-efficiency issues in the core and energy-delivery subsystems independently. This dissertation studies the application of stochastic computing in ULP platforms operating in the subthreshold regime, and includes the efficiency issues present in the energy-delivery subsystem.

The major contributions of this dissertation can be summarized as follows:

1. Stochastic computing is applied to subthreshold ULP applications. The new design space is characterized by its sensitivity to PVT variations, and energy frugality.

2. The energy and robustness benefits of stochastic computing at MEOP is quantified through analysis, simulations, and measurements from a 45-nm prototype ECG processing IC which is 4.7× more energy-efficient than the state-of-the-art and 16× more robust to voltage variations.

3. The energy optimization of compute cores and DC-DC converter in ULP platforms is conducted jointly. Core architectural techniques are proposed to improve the system (core and converter) energy efficiency.

4. A novel stochastic computing technique (*likelihood processing*) which improves on the energy benefits and robustness of existing stochastic techniques.

5. A unified framework for stochastic computation is introduced consisting of an error statistics characterization methodology, and diversity techniques to generate favorable error statistics.

1.4 Dissertation Organization

This dissertation is organized as follows:

- **Chapter 2** addresses stochastic computing in the subthreshold regime. It employs algorithmic noise-tolerance for a subthreshold filtering application to reduce the energy at the MEOP. In addition, it studies the design trade-offs between correction overhead, robustness, and energy savings in subthreshold by employing architectural and circuit-level simulations in the presence of voltage and process variations.

- **Chapter 3** describes the implementation and design of a subthreshold stochastic computing-based ECG processor in an IBM 45-nm CMOS process. Furthermore, it

demonstrates the energy efficiency and robustness benefits of stochastic computing in ULP applications through IC measurement results, and illustrates the superiority of the prototype IC compared with state-of-the-art systems in terms of energy and reliability.

- **Chapter 4** studies the joint optimization of the DC-DC converter and the computational cores to improve system energy efficiency. It shows that DC-DC converter energy losses are significant in the subthreshold regime if the DC-DC converter is designed to handle a wide range of DVS or largely varying workload characteristics. It presents core-architecture techniques to alleviate the DC-DC losses and improve the overall system energy efficiency.

- **Chapter 5** presents a unified framework for stochastic computing and introduces a new technique, likelihood processing (LP), which generates reliability information on each output bit. Design trade-offs involved in LP implementation are studied. Energy and robustness benefits in the design of a 45-nm 2D-DCT codec is quantified.

- **Chapter 6** proposes a statistical framework for error characterization at the system level and introduces design techniques to generate favorable error statistics in order to enable next-generation stochastic computing techniques. It studies the factors (PVT corner, input statistics, architecture, etc.) affecting error statistics at the system level, and presents a generalized one-time offline statistical error characterization methodology. In addition, the chapter presents design diversity techniques to ensure independent errors across the observations, and consequently improve the robustness of next-generation stochastic computing techniques. Various DSP blocks, such as adders and filters, are employed to validate the proposed error model and its characterization in a 45-nm CMOS process.

- **Chapter 7** provides a summary of contributions, concludes the work completed for this dissertation, and discusses its broader impact and future extensions.

CHAPTER 2

ENERGY-EFFICIENT AND ROBUST ULP KERNELS VIA STOCHASTIC COMPUTATION

This chapter studies the application of *algorithmic noise-tolerance* (ANT) [70] to designs operating at the minimum energy operating point (MEOP). ULP platforms for portable medical and health monitoring application, distributed wireless sensor networks, and active RFIDs operate at or around the MEOP. The subthreshold design space under stochastic computing is explored and the energy efficiency and robustness benefits are illustrated through analysis, architectural, and circuit-level simulations in a 45-nm CMOS technology.

The MEOP is defined via the tuple $(V_{dd,opt}, f_{opt}, E_{min})$ (see conventional MEOP ($MEOP_C$)) in Fig. 2.1), where E_{min}, $V_{dd,opt}$, and f_{opt} are the minimum achievable energy, and the corresponding supply voltage and frequency of operation, respectively, where E_{min} is achieved. Conventional MEOP design is well-studied [23] [26] [25] and is primarily a circuit/architectral endeavour. However, it assumes worst-case PVT variations, leading to significant energy overhead.

Stochastic computing techniques based on statistical error-compensation such as ANT has been effective in reducing energy in digital signal processing (DSP) kernels by permitting *voltage overscaling* (VOS). However, ANT and other error-resiliency techniques [53, 55, 66, 68, 69, 72] have not been applied in the subthreshold regime. This chapter studies the impact of ANT on the energy consumption E_{min} at the MEOP for an 8-tap FIR filter. The ability of ANT to cope with large pre-correction error rates p_η is exploited, and its benefits are extended to the subthreshold regime. Furthermore, the various design factors affecting robustness and energy efficiency are studied.

In the subthreshold regime, leakage energy is significant, and thus frequency overscaling

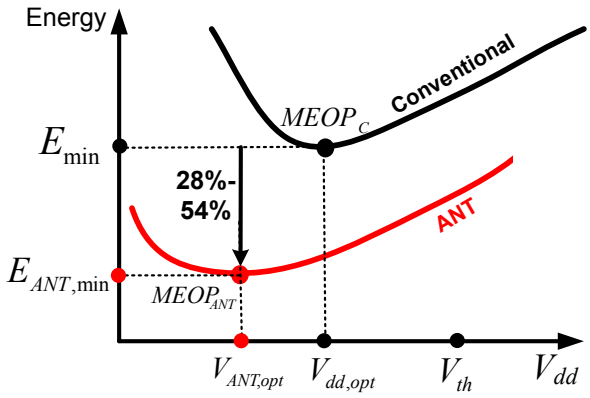


Figure 2.1: Energy in the subthreshold regime of a conventional and ANT-based designs.

(FOS), where the operating frequency is increased beyond its critical value, can be employed as well to reduce energy. In this chapter, we allow both frequency and voltage to be overscaled simultaneously,and thus achieving higher energy savings. In addition, we analyze the results in two IBM 45-nm CMOS processes with different threshold voltages, V_{th}: a high V_{th} (HVT) process, and a low V_{th} (LVT) process. We also study the energy benefits of ANT in presence of process variations. The use of two different threshold voltage changes the contribution of leakage energy at the MEOP and the delay sensitivity to voltage variations, thus affecting energy efficiency and robustness. The results of this chapter (see Fig. 2.1) can be summarized as follows:

1. Algorithmic noise-tolerance results in a new MEOP ($MEOP_{ANT}$ in Fig. 2.1) where E_{min} is reduced by up to 58% and 10%, while the operating frequency is increased by 2.5× and 1.2× in the 45-nm LVT and HVT processes, respectively, at a pre-correction error rate of 85%.

2. Under process variations in LVT process, ANT reduces E_{min} by 54% on average while maintaining a parametric yield of 99.7%.

This chapter is organized as follows: Section 2.1 analyzes the MEOP and its characterization. Section 2.2 studies the effectiveness of ANT in the subthreshold regime and and

analyzes its impact on the MEOP. Section 2.3 presents simulation results for an FIR filter in a 45-nm IBM CMOS processes which demonstrate the impact of ANT on the MEOP in the presence of voltage and process variations.

2.1 The Minimum-Energy Operating Point (MEOP)

The dominant sources of energy consumption in subthreshold are *dynamic* energy, (E_{dyn}), and the *leakage* energy (E_{lkg}), given by:

$$\begin{aligned} E_o &= E_{dyn} + E_{lkg} \\ E_{dyn} &= \alpha N C V_{dd}^2 \\ E_{lkg} &= \frac{N I_{OFF} V_{dd}}{f} \end{aligned} \tag{2.1}$$

where α is the switching activity factor, N is the number of gates each with an output load capacitance C, f is the operating frequency, V_{dd} is the supply voltage, and I_{OFF} is the OFF-state leakage current. The subthreshold current [26] as a function of gate-to-source and drain-to-source voltage is given by:

$$I_{SUB}\left(V_{GS}, V_{DS}\right) = I_o 10^{\frac{V_{GS}-V_{th}-\gamma V_{DS}}{S}} \left(1 - e^{\frac{V_{DS}}{V_T}}\right) \tag{2.2}$$

where I_o is a reference current and is proportional to the transistor W/L ratio, $S = mV_T$ is the swing factor, γ is the DIBL coefficient, V_{th} is the threshold voltage, and V_T is the thermal voltage. Using (2.2), the ON-state and OFF-state currents for an NMOS transistor are $I_{ON} = I_{SUB}\left(V_{dd}, V_{dd}\right)$ and $I_{OFF} = I_{SUB}\left(0, V_{dd}\right)$, respectively.

Assuming the critical path of the computational kernel has a logic depth of L gates each

with an output load capacitance C, the operating frequency f is given by:

$$f = \frac{I_{ON}}{\beta LCV_{dd}} \tag{2.3}$$

where β is a fitting parameter needed to match the finite signal rise and fall times. The subthreshold frequency of operation decreases exponentially with V_{dd} reduction due to the exponential dependance of I_{ON} on V_{dd} in (2.2). This leads to an exponential increase in *leakage* energy as seen by substituting (2.3) in (4.1) to get:

$$E_{lkg} = \beta NLCV_{dd}^2 \frac{I_{OFF}}{I_{ON}} = \beta NLCV_{dd}^2 10^{\frac{-V_{dd}}{S}} \tag{2.4}$$

and the total subthreshold energy is given by:

$$E_o = \alpha NCV_{dd}^2 + \beta NLCV_{dd}^2 10^{\frac{-\gamma V_{dd}}{S}} \tag{2.5}$$

Therefore, reducing V_{dd} in the subthreshold region decreases E_{dyn} but increases E_{lkg} exponentially so that operating point $MEOP_C$ in Fig. 2.1 exists.

2.2 MEOP via ANT

In this section, we study the energy behavior of ANT in the subthreshold regime and study its impact on the MEOP. For ANT to provide energy efficiency, it is necessary that the errors in the main block (see Fig. 1.7(a)) be primarily due to enhancement of its energy efficiency. These properties can be satisfied when errors in the main block are induced by either VOS, FOS, or a combination of both since leakage energy contributes significantly to total energy in subthreshold.

In VOS, the supply voltage is reduced below the critical voltage $V_{dd,crit}$ needed for error-free operation while keeping the operating frequency fixed, i.e., $V_{dd} = K_{VOS}V_{dd,crit}$ and $f = f_{crit}$

where $K_{VOS} < 1$ is the VOS factor. In FOS, V_{dd} is kept fixed while f is increased beyond f_{crit}, i.e., $V_{dd} = V_{dd,crit}$ and $f = K_{FOS} f_{crit}$ where $K_{FOS} > 1$ is the FOS factor. FOS not only reduces leakage energy due to a smaller operating clock period, but also enables higher performance (frequency). As most arithmetic computations are least-significant-bit (LSB) first, timing violations due to VOS or FOS are generally large magnitude most-significant-bit (MSB) errors. Thus, timing violations due to VOS or FOS satisfy the error distribution shown in Fig. 1.7(b).

Given an application-level error-tolerance limit, e.g., maximum SNR loss, VOS and FOS can be applied simultaneously to save energy. The estimator is designed such that it operates error-free due to its lower complexity compared to the main block, and the ANT residual (post-correction) error is within the application tolerance limit. The energy of a subthreshold ANT system E_{ANT} is given by:

$$\begin{aligned} E_{ANT} = & \quad K_{VOS}^2 \left(1 + \tfrac{\alpha_{est} N_{est}}{\alpha N}\right) E_{o,dyn} + \\ & \tfrac{1}{K_{FOS}} K_{VOS} \left(1 + \tfrac{N_{est}}{N}\right) \tfrac{I_{OFF,K_{VOS} V_{dd,crit}}}{I_{OFF,V_{dd,crit}}} E_{o,lkg} \end{aligned} \tag{2.6}$$

where N_{est} is the number of additional gates needed to implement the estimator and the decision block (ANT overhead), and α_{est} is the average switching activity of the N_{est} nodes. Note that, a large class of estimators in ANT operates on the high-order bits of the input, which usually have a lower switching activity factor than the low-order bits, and thus $\alpha_{est} < \alpha$. Several design factors, such as the hardware error rate of the main block (p_η), the application error tolerance, and the estimator complexity, determine the total system energy consumption. It will be shown that a new MEOP, $MEOP_{ANT}$ characterized by the tuple $(V_{ANT,opt}, f_{ANT,opt}, E_{ANT,min})$, exists, where $V_{ANT,opt} \leq V_{dd,opt}$, $f_{ANT,opt} > f_{opt}$, and $E_{ANT,min} < E_{min}$. The MEOP energy $E_{ANT,min}$ depends on the estimator overhead, the application error tolerance, and delay sensitivity to voltage variations (process threshold

voltage). Next, we illustrate the energy savings and the different trade-offs involved in subthreshold ANT using an FIR filter.

2.3 Simulations and Results

We design a 23-b output FIR filter (see Fig. 2.2(a)), which is a widely-used DSP kernel in the subthreshold regime. The FIR filter operates at an error-free critical supply voltage and frequency ($V_{dd,crit}$, f_{crit}), and computes $y[n] = \sum_{i=0}^{7} x[n-i] \times h_i$ where $x[n]$ is a 10-b input signal, h_i's are the 10-b filter coefficients, and n is the clock-cycle/time index. The filter uses a ripple-carry adder-based architecture as a building block for the adders and multipliers, and achieves an SNR of 17 dB when operating hardware error-free. The filter is designed in the IBM 45-nm LVT and HVT CMOS processes. The following simulation procedure is employed to study the impact of ANT on the MEOP.

2.3.1 Simulation Procedure

Given an application error-tolerance limit (maximum SNR loss), energy/design margins are reduced so that the ANT main block has a specific pre-correction error rate (p_η) which can be achieved by either VOS at K_{VOS} or FOS at K_{FOS} or a combination of both. To compute p_η and system energy, and study the effect of timing errors on application metric (SNR), the simulation procedure is as follows:

1. Circuit Simulations: we employ circuit simulations using HSPICE in the 45-nm IBM CMOS process in order to characterize the worst-case delay and power of a limited-size gate library (1-bit adder, and-gate, or-gate, inverter, etc.) at different voltages or PVT corners. We use the analytical models presented in (2.3) and (2.5) to fit the delay and power numbers obtained for each gate.

2. Gate-Level Netlist Simulations: we employ an RTL-level structural Verilog model of the

filter to generate the erroneous output $y[n]$, using individual gate delays obtained from step 1 at $V_{dd} = K_{VOS}V_{dd,crit}$ ($V_{dd} = V_{dd,crit}$), and operating at $f = f_{crit}$ ($f = K_{FOS}f_{crit}$) for VOS (FOS).

3. We determine the main block error rate p_η and the SNR of the filter under timing errors by comparing the erroneous RTL output to the error-free output.

4. Power Estimation: we compute the overall system leakage and dynamic energy/power by summing up the leakage and dynamic power estimates of the filter constituent gates obtained from step 1, while taking into consideration the average activity factor α of each gate in step 2.

2.3.2 MEOP of (Error-Free) Filter

We employ steps 1 and 4 of the simulation procedure to estimate the energy consumption of the conventional error-free FIR filter. We validate the subthreshold energy consumption by comparing the estimated values to HSPICE simulations of the complete FIR filter with an average switching activity factor $\alpha = 0.1$ in the HVT and LVT 45-nm IBM CMOS processes. Figures 2.2(b) and (c) show the analytical model estimates of the filter energy and operating frequency approximate SPICE simulations very well. Several observations can be made when comparing the energy and frequency behavior of the filter in LVT to that in HVT process and all can be attributed to the difference in V_{th} between the two processes:

1. E_{lkg} in LVT process is significantly larger (20× in near/superthreshold) than E_{lkg} in HVT process while E_{dyn} is almost the same in the two processes.

2. The total filter energy in LVT process is dominated by leakage even for near/superthreshold supply voltages ($E_{lkg} \approx 4E_{dyn}$), while E_{lkg} in HVT starts to dominate total energy only in the subthreshold regime.

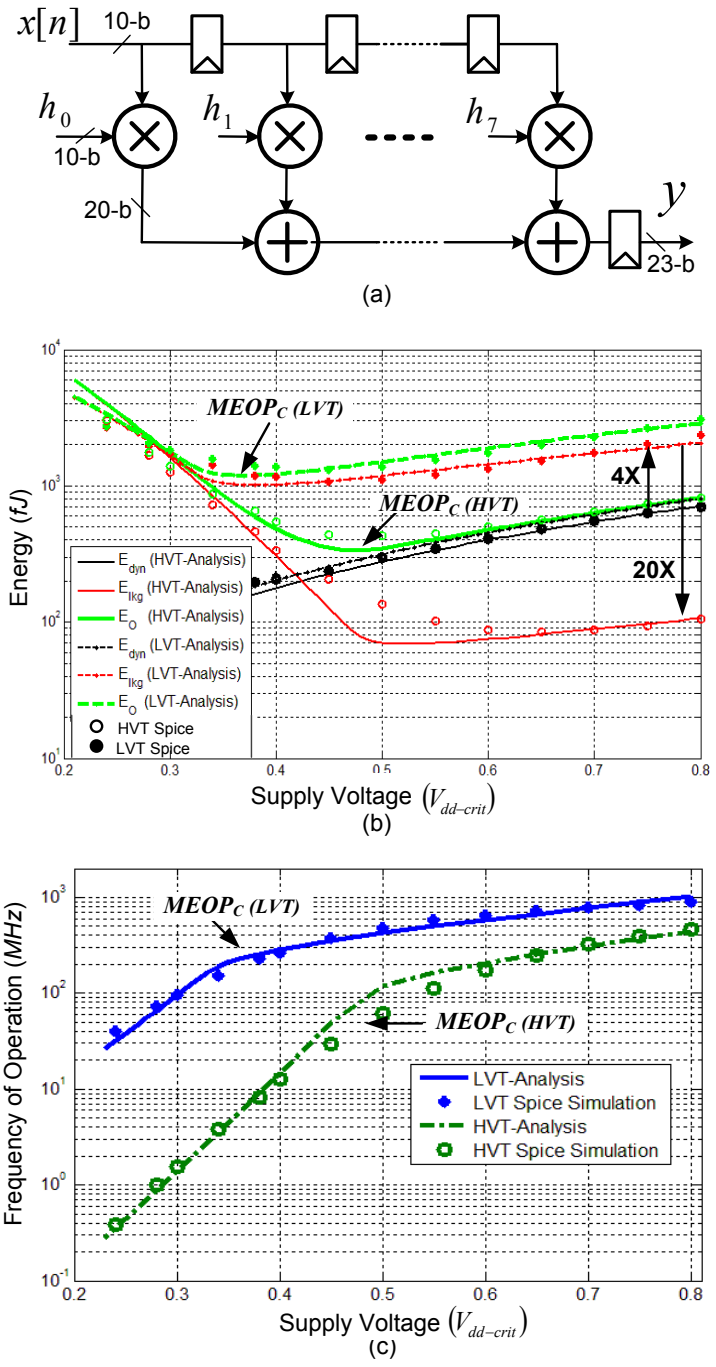


Figure 2.2: Validation of energy and frequency models for an eight-tap FIR filter in 45-nm CMOS processes: (a) direct-form architecture, (b) energy model, and (c) throughput model.

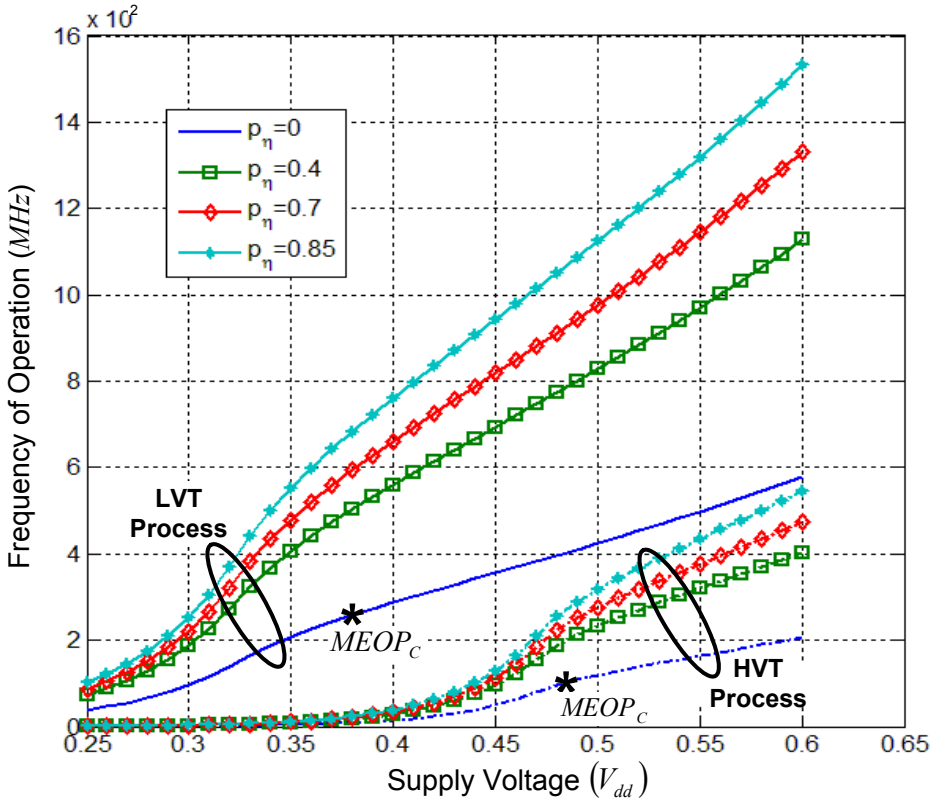


Figure 2.3: Voltage-frequency plane of iso-p_η curves in the 45-nm LVT (solid lines) and HVT (dotted lines) processes.

3. The filter in LVT achieves higher operating frequency than that in HVT. The filter operating frequency starts to decrease significantly for supply voltages less than 0.38 V in LVT and 0.5 V in HVT process, and the corresponding E_{lkg} starts to increase at similar voltages in each process.

4. The conventional MEOP $MEOP_C$ in LVT process is reached at ($V_{dd,opt} = 0.38\,\text{V}$, $f_{opt} = 240\,\text{MHz}$, $E_{min} = 1022\,\text{fJ}$) while that of HVT is reached at ($V_{dd,opt} = 0.48\,\text{V}$, $f_{opt} = 80\,\text{MHz}$, $E_{min} = 335\,\text{fJ}$).

2.3.3 Energy vs. Error Rate and Error Rate vs. SNR Characterization

In this section, we characterize the voltage-frequency operating points (V, f) required to achieve a given pre-correction error rate p_η in the filter main block and compare the effect of FOS vs. VOS on p_η. The simulation procedure in Section 2.3.1 is employed to characterize the (V, f) iso-p_η curves in the 45-nm LVT and HVT processes (see Fig. 2.3). Note that a horizontal translation (fixed frequency) in Fig. 2.3 corresponds to VOS , a vertical translation (fixed voltage) corresponds to FOS, and an arbitrary translation corresponds to simultaneous VOS and FOS. We can see that as the critical supply voltage reduces, the horizontal and vertical gaps between different p_η curves reduce due to the increased sensitivity of delay as supply voltage approaches the threshold voltage. Moreover, the HVT process has larger delay sensitivity as compared to the LVT process due to higher threshold voltage. To further compare FOS and VOS at $MEOP_C$, Figs. 2.4 (a) and (b) show the raw hardware error rate, p_η, at the output of the main block as well as its energy consumption under VOS and FOS in the LVT and HVT processes. Under FOS, p_η is the same for the LVT and HVT processes since errors due to FOS are only a function of the architecture and are independent of the underlying circuit and process fabric. However, under VOS, the LVT process has a lower p_η than HVT for same K_{VOS} since its $|V_{dd,opt} - V_{th}|$ at $MEOP_C$ is less than that at the $MEOP_C$ of the HVT process, which leads to lower percentage increase in delay at C-MEOP due to VOS in the LVT process. This can be seen in Fig. 2.2(c) where the rate of change (slope) of frequency at the C-MEOP in the LVT process is less than that in the HVT process. Moreover, note that FOS is more robust than VOS at a given p_η as seen in Fig. 2.4(a) by comparing the slope of the respective p_η curves. Small variations in K_{VOS} lead to large variations in p_η unlike variations in K_{FOS}. This is due to the exponential relation between voltage and delay in the subthreshold regime.

The percentage energy savings in E_{min} at the $MEOP_C$ in the LVT and HVT processes due to VOS is equal for the same K_{VOS} (see Fig. 2.4(b)) since percentage energy savings

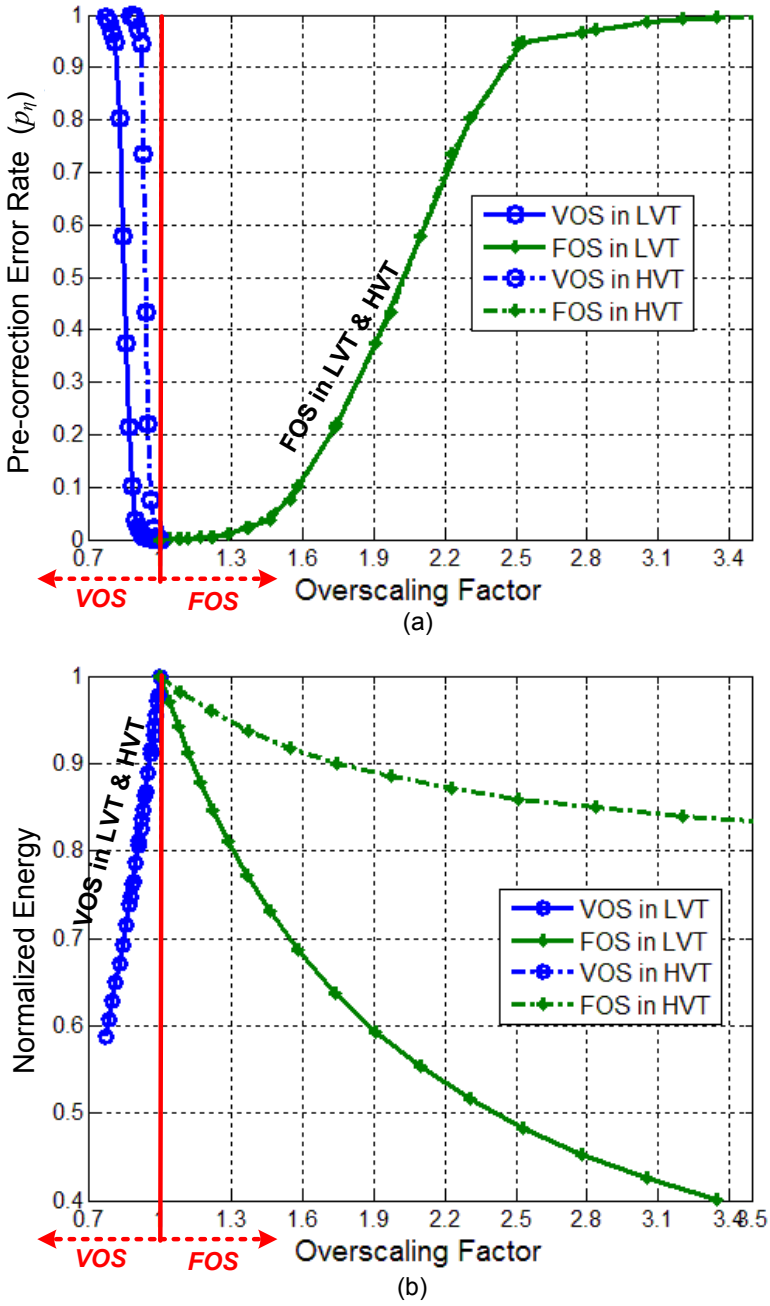


Figure 2.4: Pre-correction error rate and energy characterization of the 8-tap FIR filter under VOS (x axis ≤ 1) and FOS (x axis ≥ 1): a) error rate p_η vs. K_{VOS} and K_{FOS}, and b) normalized energy vs. K_{VOS} and K_{FOS} (error-compensation overhead is not included).

depends on K_{VOS} and is independent of supply or threshold voltage, unlike the absolute energy. However, under FOS, the percentage energy savings is larger for the LVT process than the HVT process for the same K_{FOS} since total energy in the LVT process at $MEOP_C$ is dominated by E_{lkg} compared to the HVT process (as seen in Fig. 2.2(b) and discussed in Section 2.3.2) and FOS reduces only E_{lkg}.

To see the effect of ANT on the application performance metric (SNR), we employ a reduced precision-redundancy (RPR) version of the main block filter as an estimator [79], as shown in Fig. 2.5(a). The main filter output precision is 23-b with 10-b input and coefficients, and its architecture is shown in Fig. 2.2(a), while the RPR estimator block has similar architecture to the main filter while processing only the B_e most-significant bits of the main filter 10-b inputs and coefficients ($B_e < 10$) to generate an estimated output having $2B_e+3$ bits. ANT estimation and correction circuits are operated at the same voltage and frequency as the main block. However, they do not have the same timing error rates as the main block due to their reduced complexity. In this case, the estimator has lower precisions than the main block, resulting in longer timing slack. We follow the simulation procedure outlined previously in Section 2.3.1 to estimate the SNR under hardware errors. Figure 2.5(b) shows the SNR of the uncorrected (conventional) filter and that of the ANT filter with different B_e. The conventional filter SNR drops catastrophically as p_η increases above 0.1% while the SNR of the ANT filter remains within 0.8 dB of the error-free output for p_η values up to 70% for $B_{est} = 5$ (point **B**). Higher-precision estimators reduce the residual error at the output resulting in an SNR drop of less than 0.2 dB (point **A**) . However, they operate error-free at lower p_η (40%) than the lower precision estimators due to their increased critical path. Next, we study the energy consumption of the ANT filter while accounting for both estimator and main block energy.

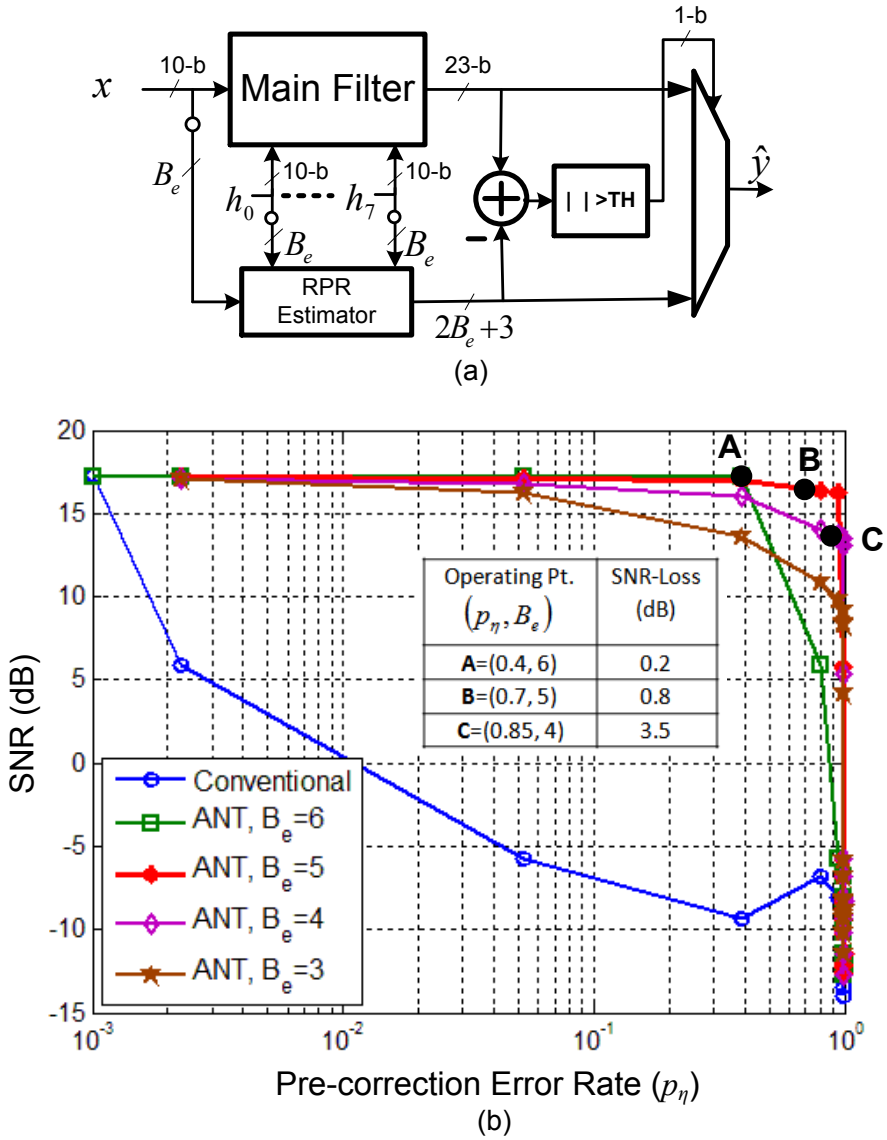


Figure 2.5: SNR vs. error rate for the 8-tap reduced precision-redundancy (RPR) ANT-based filter with different estimator precisions (B_e): (a) architecture, and (b) SNR performance vs. p_η.

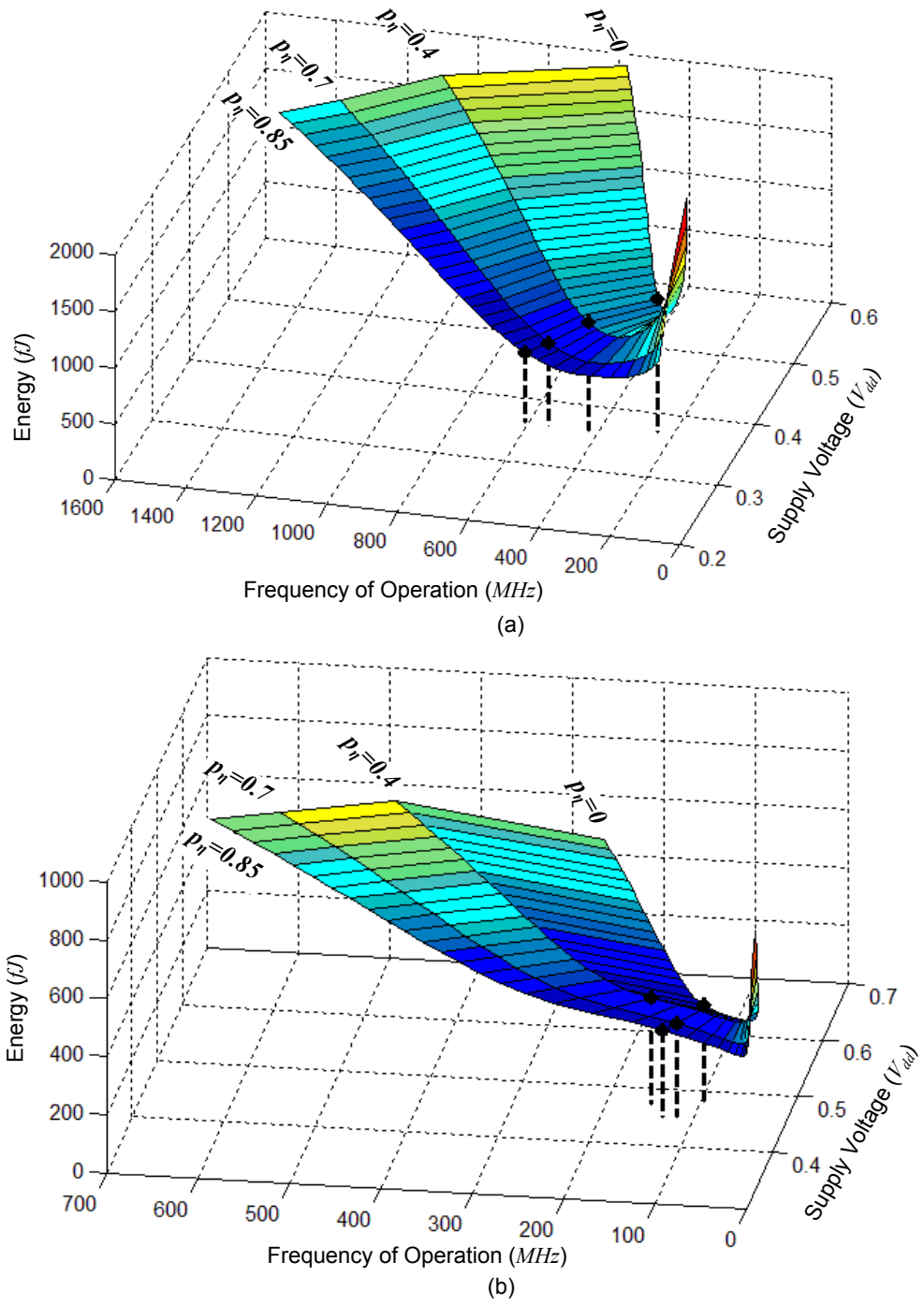


Figure 2.6: Energy of ANT FIR filter (including error-compensation overhead for $p_\eta \neq 0$) at different pre-correction error rates and estimation overhead: (a) the 45-nm LVT process, and (b) the 45-nm HVT process.

Table 2.1: MEOP comparison of conventional and ANT filters in the 45-nm LVT process.

Design Type	SNR (dB)	$V_{dd,opt}$	f_{opt}(MHz)	E_{min}(fJ)	Energy Savings w.r.t Conventional 0 (17.1 dB)	Energy Savings w.r.t Conventional at same SNR
Conventional 0 ($p_\eta = 0$)	17.1	0.38	240	1022	0%	
Conventional 1 ($p_\eta = 0$)	16.9	0.375	240	998	2.3%	
Conventional 2 ($p_\eta = 0$)	16.3	0.371	240	946	7.4%	
Conventional 3 ($p_\eta = 0$)	13.6	0.370	240	891	12.8%	
ANT ($p_\eta = 0.4, B_e = 6$)	16.9	0.36	430	957	20%	4.1%
ANT ($p_\eta = 0.7, B_e = 5$)	16.3	0.36	542	738	38%	22%
ANT ($p_\eta = 0.85, B_e = 4$)	13.6	0.36	610	632	47%	29%

2.3.4 MEOP of ANT Filter at Nominal Process Corner

Given an application error tolerance (e.g. maximum allowable SNR loss), the optimal ANT configuration (p_η, B_{est}) can be determined from Fig. 2.5. For example, for an SNR-loss of 0 dB, 0.2 dB, 0.8 dB, and 3.5 dB, the optimal ANT configurations are **Conventional**, **A**, **B**, and **C**, respectively, where the pre-correction error rates p_η are 0, 0.4, 0.7, and 0.85 and estimator precisions B_{est} are 0-b, 6-b, 5-b and 4-b, respectively. Figure 2.3 shows the corresponding voltage-frequency pair needed to achieve the required p_η at the corresponding optimal ANT configurations. The total system energy behavior, including ANT estimation and correction overhead when $p_\eta \neq 0$, is shown in Fig. 2.6. V_{dd} and f at the MEOP are different in all configurations indicating that the system needs to be operated differently for each configuration. The MEOP of each configuration is shown in Tables 2.1 and 2.2 for the LVT and HVT processes, respectively. ANT achieves up to 38% and 47% energy saving at

Table 2.2: MEOP comparison of conventional and ANT filters in the 45-nm HVT process.

Design Type	SNR (dB)	$V_{dd,opt}$	f_{opt}(MHz)	E_{min}(fJ)	Energy Savings w.r.t. Conventional 0 (17.1 dB)	Energy Savings w.r.t. Conventional at same SNR
Conventional 0 ($p_\eta = 0$)	17.1	0.48	80	335	0%	N.A.
Conventional 1 ($p_\eta = 0$)	16.9	0.478	80	329	1.8%	N.A.
Conventional 2 ($p_\eta = 0$)	16.3	0.475	80	324	3.3%	N.A.
Conventional 3 ($p_\eta = 0$)	13.6	0.47	80	311	7.2%	N.A.
ANT ($p_\eta = 0.4, B_e = 6$)	16.9	0.47	141	369	−11%	−12.1%
ANT ($p_\eta = 0.7, B_e = 5$)	16.3	0.45	107	326	2.4%	−0.6%
ANT ($p_\eta = 0.85, B_e = 4$)	13.6	0.45	122	299	10%	3.9%

error rates of $p_\eta = 0.7$ and 0.85, respectively, in the LVT process while incurring an SNR loss of 0.8 to 3.5 dB when compared to an error-free conventional design at 17.1 dB.

The SNR loss incurred by ANT filters can be traded in conventional design for lower energy by reducing input and coefficient precisions (see Conventional 1, 2, and 3 in Table 2.1). Comparing conventional designs to ANT filters at the same SNR, the energy savings of ANT ranges from 4% to 29%. From Table 2.1, we can also see that MEOP ANT not only operates at lower supply voltage $V_{dd,opt}$ but also provides increased frequency of operation reaching 1.8× and 2.25× at $p_\eta = 0.7$ and 0.85, respectively in the LVT process. In the HVT process, leakage energy is less dominant at MEOP compared to the LVT process, and the error rate shows greater sensitivity to VOS as illustrated in Fig. 2.4(a). That is why the energy benefits of ANT compared to the conventional system is less pronounced in the HVT process where at most 10% energy savings are achieved under ANT with an SNR-loss of 3.5 dB (see Table 2.2). Note that at $p_\eta = 0.4$ in the HVT process, ANT results in 11%

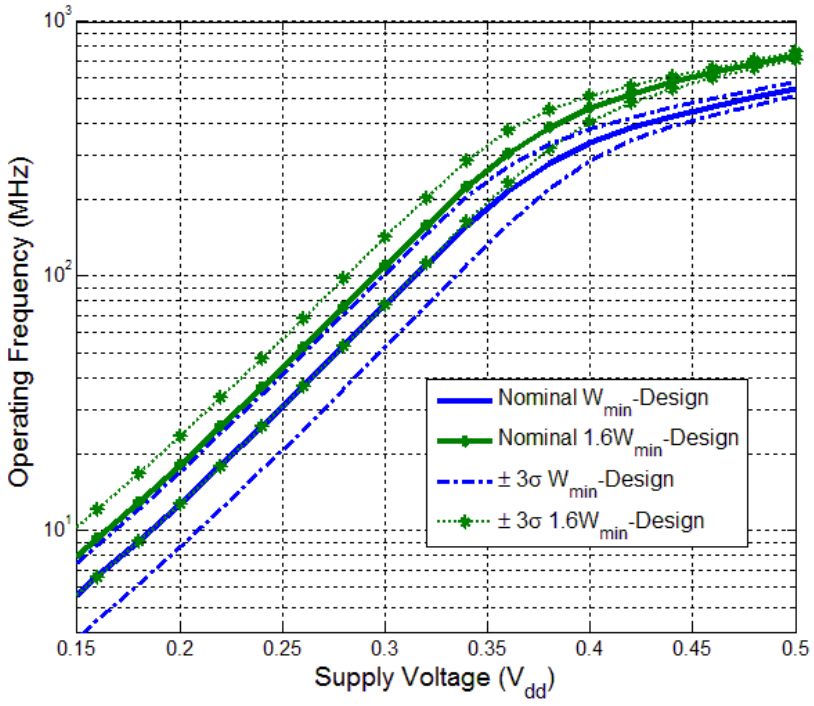


Figure 2.7: FIR filter frequency fluctuations under process variations using minimum-size (W_{min}) and 1.6-W_{min} transistors in the 45-nm LVT process.

energy overhead, since error correction overhead offsets the energy savings obtained by VOS and FOS.

An important factor to consider in Fig. 2.6 is that the energy curves under ANT are flatter than those of conventional error-free design, indicating that ANT designs are less sensitive to V_{dd} variations. All this clearly shows that statistical error compensation saves considerable energy and enhances robustness in energy-constrained subthreshold applications.

2.3.5 MEOP of ANT Filter Under Process Variations

Another design challenge, especially in MEOP designs, is process variations. Timing errors induced by process variations severely affect the filter SNR. The traditional design philosophy will reduce design margins to guarantee application requirements and performance at the

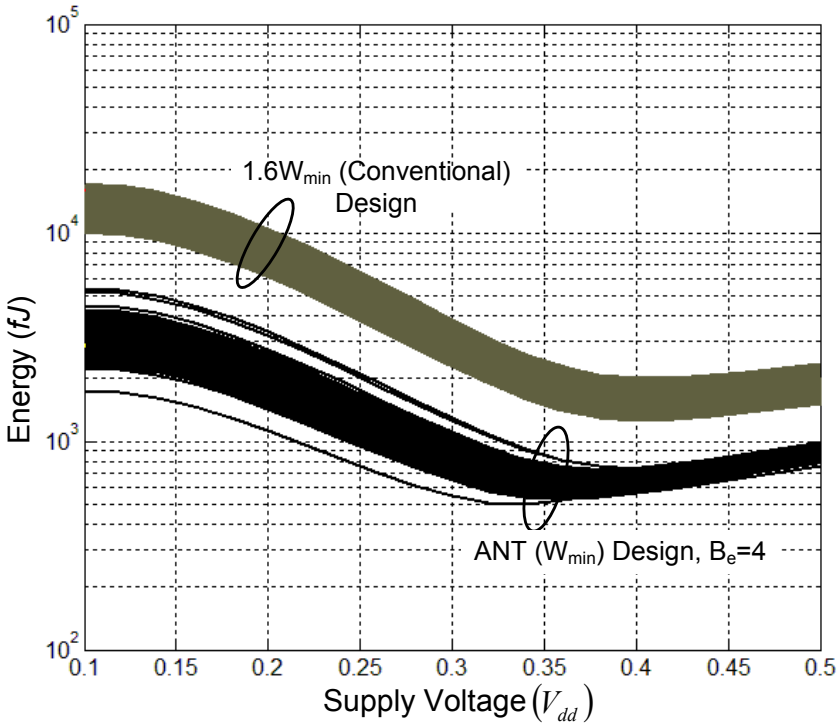


Figure 2.8: Energy under process variations for the 8-tap FIR filter using up-sized (1.6-W_{min}) design and minimum-sized (W_{min}) ANT design

worst-case process corner. A large portion of within-die (WID) variations are due to random dopant fluctuations (RDF) which cause large variations in threshold voltage [80]. Increasing the transistor sizes will reduce RDF at the expense of an increased energy consumption. Using minimum-size (W_{min}) transistors will guarantee the lowest energy consumption at MEOP but will incur a loss of yield if the nominal/target performance is not met.

To simulate the effect of process variation on performance and energy, the delay distribution of various gates used in the filter were obtained via Monte Carlo simulations in the LVT 45-nm IBM CMOS process with WID variations enabled. These delay distributions are sampled to obtain different instances of the filter. These instances are then simulated at the RTL-level to find the error-free operating frequency of each filter instance. Figure 2.7 shows the frequency distribution of the filter under process variations at different supply voltages

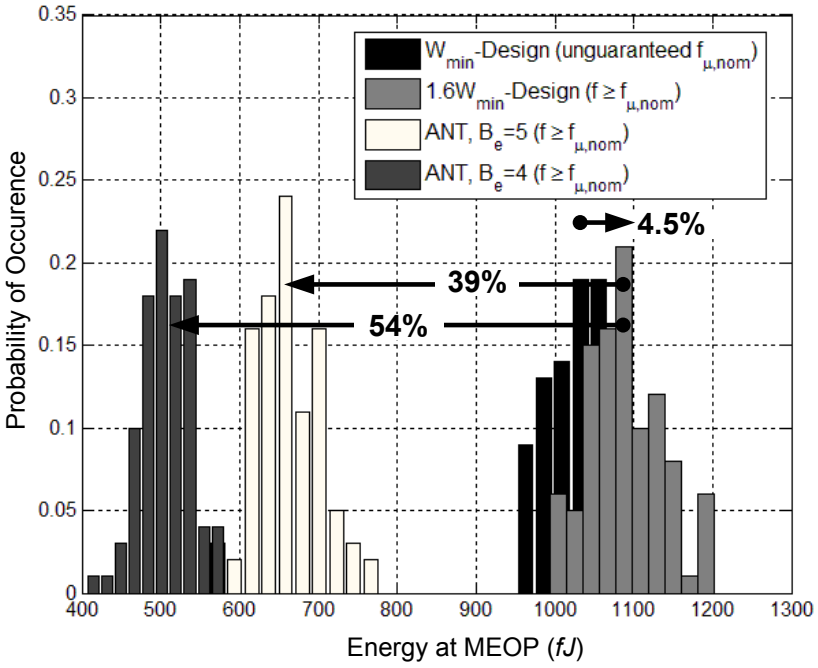


Figure 2.9: Energy distributions at the MEOP of the minimum-sized (nominal) design, up-sized design, and ANT minimum-sized designs with $B_e = 4$ and 5 (error compensation overhead is included).

and transistor widths. If we need to guarantee an operating frequency equals the nominal operating frequency of the minimum-sized design $f_{\mu,nom} = 240\,\text{kHz}$ at MEOP under WID process variations, the transistor sizes will have to be increased by at least 60% to maintain a constant parametric yield of 99.7%.

Using HSPICE power estimates for each constituent gate of each filter instance and taking into consideration the switching activity factor of each gate, the energies of the up-sized (1.6-W_{min}) conventional design and the minimum-sized ANT design including error compensation overhead are shown in Fig. 2.8, and the corresponding energy distributions at MEOP are shown in Fig. 2.9. On average, there is a 4.5% increase in energy to guarantee an operation frequency of $f_{\mu,nom}$ under process variations in a conventional design. On the other hand, a minimum-sized design, which employs ANT with FOS in order to meet

throughput and correct for timing violations, achieves a mean energy savings of 39% and 54% when $B_e = 5$ and $B_e = 4$, respectively (see Fig. 2.9). These results indicate the benefits of stochastic computing in saving energy in the subthreshold regime while guaranteeing a desired parametric yield.

2.4 Summary

In this chapter, the impact of stochastic computing on MEOP was studied. An ANT FIR filter was employed as a test case to demonstrate significant energy savings and robustness under process and voltage variations. This work shows that, similarly to the superthreshold regime, stochastic computing designs provide robustness and energy benefits in the subthreshold regime over energy-optimal error-free designs. These benefits are attributed to the ability of stochastic computing techniques to cope with relatively high error rates.

CHAPTER 3

A 14.5 FJ/CYCLE/K-GATE, 0.33 V STOCHASTIC COMPUTING-BASED ECG PROCESSOR IN A 45-NM CMOS

Chapter 2 demonstrated through analysis, RTL, and circuit-level simulations, the application of stochastic computing at the MEOP in order to allow the design of energy-efficient robust ULP platforms and ICs. This chapter describes the design and implementation of a ULP stochastic-computing prototype IC for electrocardiogram (ECG) analysis, and illustrates the energy and robustness benefits of stochastic computing at MEOP through measurements.

Spiralling health care costs and a rapidly aging population have lead to a growing interest in personal and preventive health care systems and telemedicine [81]. Real-time monitoring and analysis of ECGs is expected to have a significant impact on personal and preventive health care by enabling expert intervention in the early stages of cardiovascular diseases (CVDs) [81–83], which account for 30% of all deaths [84, 85]. Since vital biomedical signal bandwidths are less than 1 MHz, energy efficient health monitoring systems operate at or near the MEOP. Recently, several wearable/implantable biomedical devices and SoCs [29, 33–38], have been reported at or near MEOP. However, they have been designed error free assuming worst-case PVT variations, which leads to large energy overhead and reduced MEOP design margins.

The stochastic computing-based subthreshold prototype IC is designed in 45 nm IBM SOI CMOS process and implements the Pan-Tompkins algorithm (PTA) [86]. PTA is a derivative-based algorithm widely used in ECG processing systems because it does not require extensive computation, manual segmentation of data, a training phase, or patient-specific modifications and provides an acceptable beat-detection accuracy [82, 87]. Measurement results show that ANT reduces E_{min} by 28% compared to the conventional (error-free)

processors while maintaining acceptable beat-detection performance. Furthermore, ANT enables the IC supply voltage to be scaled to 15% below its critical value at MEOP, while compensating for a 58% pre-correction error rate p_η. These results represent an improvement of 19× in beat-detection performance, and 600× in p_η over conventional systems. The prototype IC consumes 14.5 fJ/cycle/1k-gate and exhibits 4.7× better energy efficiency than the state-of-the-art while tolerating 16× more voltage variations.

This chapter is organized as follows: Section 3.1 presents a brief background on PTA for ECG processing. Section 3.2 describes the architecture and implementation details of the ECG processor. Section 3.3 presents measurement results illustrating the benefits of stochastic computing in subthreshold.

3.1 The Pan-Tompkins Algorithm (PTA)

ECG consists of periodic QRS complexes (see Fig. 3.1(a)) which reflect the electrical activity in the heart during ventricular contraction. Accurate real-time QRS detection and beat-to-beat (RR) interval (see Fig. 3.1(a)) extraction are the basis for heart monitoring providing a simple noninvasive and quantitative assessment of cardiac health. Several techniques have been proposed for RR-interval extraction such as derivative-based algorithms, filter banks, wavelet transforms, neural networks, genetic algorithms, and others (see [89] and [90] for a comprehensive coverage of the different algorithms).

The PTA is the most widely used algorithm for QRS detection. It consists of noise-removal filters, derivative and squaring stage, a moving average, and a peak detector (see Fig. 3.2). The raw ECG signal x (see Fig. 3.1(a)) is corrupted by various noise artifacts such as 60 Hz noise, muscle noise, motion artifact, and skin interface [91]. PTA employs a band-pass filter as a first step to maximize the QRS signal-to-noise ratio (SNR) in the QRS frequency band of interest which is 5 Hz to 15 Hz. The band-pass filter is implemented using a cascade of a low-pass filter (LPF) with a cut-off frequency of 15 Hz and a high-pass filter

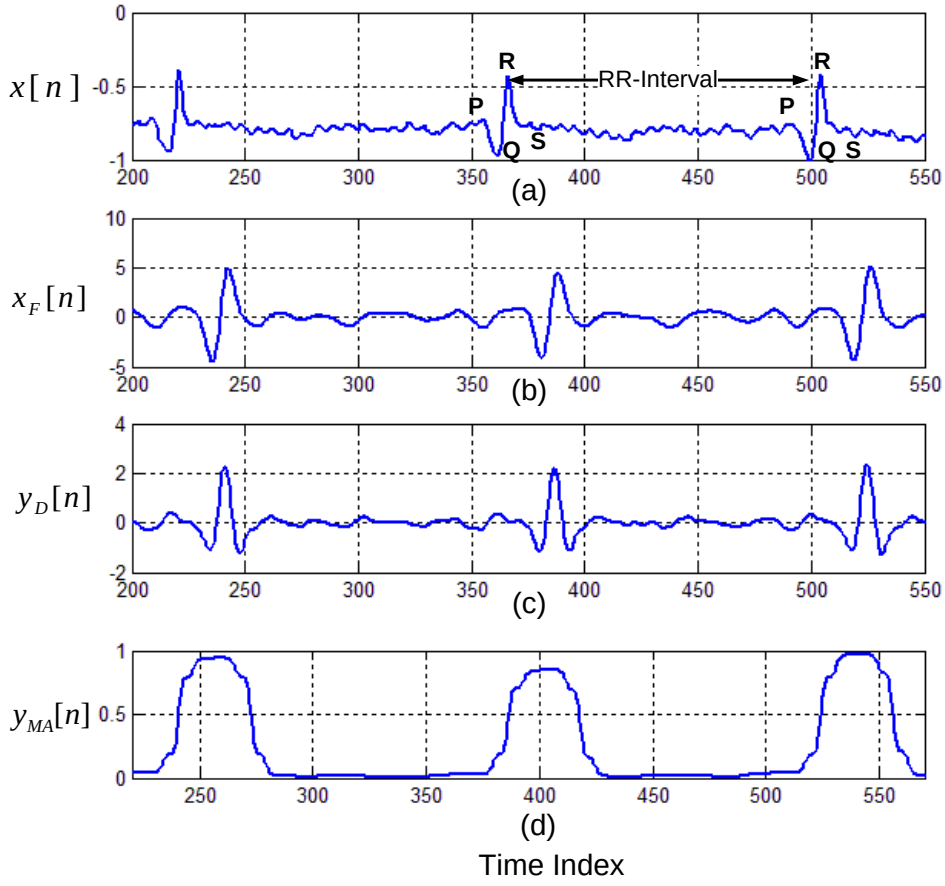


Figure 3.1: ECG processing of a segment of an MIT-BIH [88] database record: (a) input (noisy) ECG, (b) filtered ECG, (c) ECG at derivative output, and (d) ECG at moving average output.

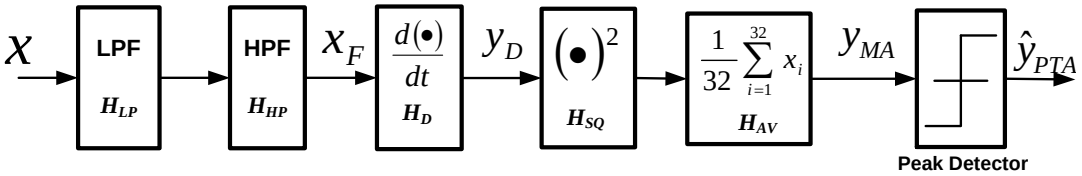


Figure 3.2: Block diagram of the Pan-Tompkins algorithm (PTA) for ECG processing

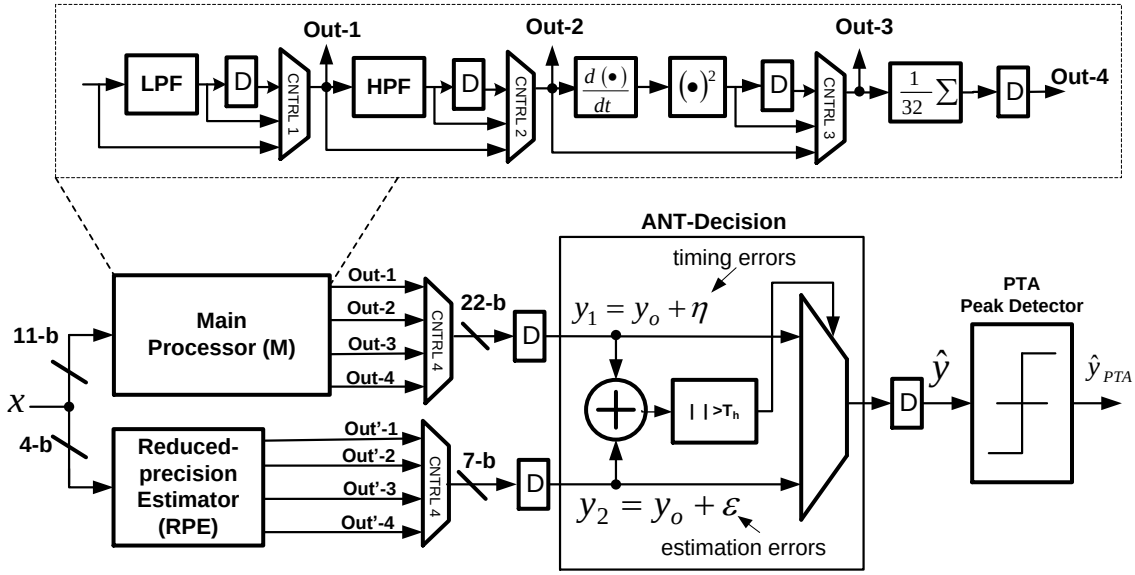


Figure 3.3: Architecture of ANT-based ECG Processor

(HPF) with a cut-off frequency of 5 Hz. The filtered ECG signal x_F is then differentiated in order to amplify the higher frequencies characteristic of ECG the wave (QRS-complex) while attenuating the lower frequencies characteristic (P and T-waves in Fig.3.1(b)). Squaring is then applied to intensify the higher frequency characteristics. The squared signal y_{SQ} is then passed through a moving window integrator to provide information about the QRS-complex width. The final stage in PTA is an adaptive detector which exploits information about the QRS amplitude, slope, and width, as well as physiological properties of ECG in order to determine the locations of the R-waves. Thus, the final output y_{PTA} is a pulse train where each pulse indicates a location of an R-wave. Irregular RR-intervals can then signal the onset of a CVD [82] [83].

3.2 Architecture and Implementation

Algorithmic noise tolerance is employed to provide robustness and reduce energy of the designed ECG process. The chip architecture is shown in Fig. 3.3. The main processor (M)

Table 3.1: Transfer function of building blocks in PTA.

Block	Transfer Function
LPF	$H_{LP}(z) = \frac{1-2z^{-6}+z^{-12}}{1-2z^{-1}+z^{-2}}$
HPF	$H_{HP}(z) = \frac{-1+32z^{-16}+z^{-32}}{1+z^{-1}}$
Derivative	$H_D(z) = \frac{1}{8}(-z^{-2} - 2z^{-1} + 2z^{1} + z^{2})$
MA	$H_{MA}(z) = \frac{1}{32}\sum_{i=0}^{31} z^{-i}$

consists of low-pass (LPF) and high-pass (HPF) filters, a derivative and squaring blocks (DS), and a moving average block (MA). The main block operates on an 11-b input ECG signal. A reduced precision redundancy is employed as estimator. The reduced precision estimator (RPE) operates on the 4-b MSB of the input and each block in M is replicated in RPE at the reduced precision. The estimator gate complexity is 32% of the main ECG processor. The ANT-decision block in Fig. 3.3 employs the principle of statistical detection in order to compute the corrected output. The PTA peak detector, similar to ANT-decision block, presents a challenge for statistical error compensation since being non-linear with one-bit output makes it difficult to design a low-complexity estimator. Thus, it is designed to operate error-free with enough timing slack to handle VOS or FOS.

A reconfigurable data path in M and RPE is employed where pipelining latches (D) can be introduced at the output of different blocks. This allows us to control the error locations in the ECG processing algorithm. The filter coefficients are designed to be a power of 2 to reduce complexity and are implemented as proposed in [86]. The transfer functions of basic blocks and the corresponding architectures are shown in Table 3.1 and Fig. 3.4, respectively. The LPF and HPF are designed using pole-zero cancelation on the unit circle in order to have integer coefficients. The derivative is a five-point derivative which approximates an ideal derivative up to 30 Hz. The moving average window size is 32 samples to accommodate the largest QRS-complex width (160 ms) at a sample rate of 200 samples/s. The moving average block (see Fig. 3.4(c)) is designed using Wallace-tree carry-save adders. For the rest of the blocks in the ECG processor, the basic computation structure employs ripple carry adders and array multiplier.

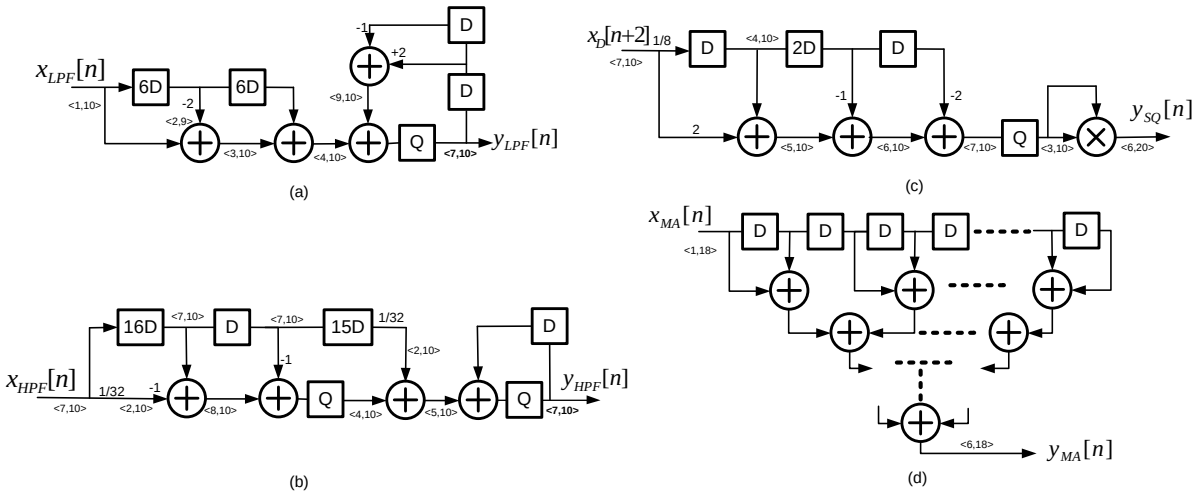


Figure 3.4: Architecture of building blocks in PTA: (a) LPF, (b) HPF, (c) derivative-square (DS) block, and (d) moving average (MA) block. The precision shown is for the main block M and the notation $< n_1, n_2 >$ represents n_1 integer-bits and n_2 floating-bits.

The chip is implemented in a regular V_{th} 45-nm 1V IBM SOI CMOS process. An ARM standard cell library is employed in synthesis. The usage of library cells is restricted to minimum strength cells to reduce energy and introduce timing slack between MSB and LSB part, and thus allow a graceful increase in error rate at the output of M when VOS or FOS is applied. A total of 12 power domains are employed, one for each of the following 6 power domains: 1) M-block filters and DS, 2) RPE-block filters and DS, 3) M-block MA, 4) RPE-block MA, 5) ANT-decision block, and 6) PTA peak detector, and separately for combinational and sequential logic within each of the six modules in order to avoid the failure of sequential logic at very low voltages.

The final design has a total of 36 k NAND2 gates, and a total chip area including the pad frame of 1.25 mm × 1.3 mm. The core area is approximately 0.7 mm × 0.7 mm. The chip microphotograph is shown in Fig. 3.5.

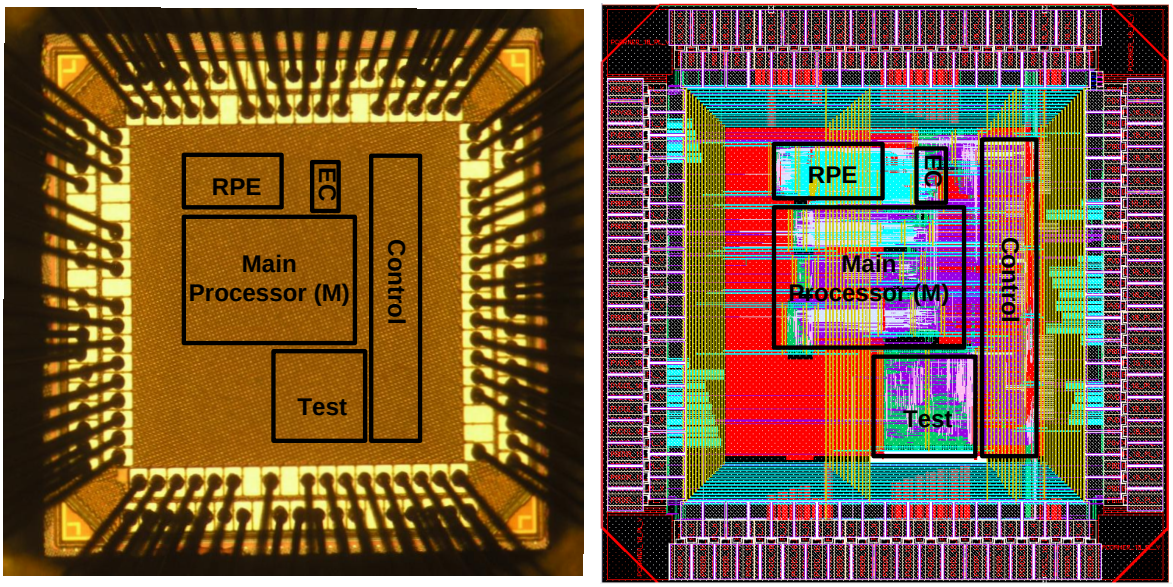


Figure 3.5: Die photo of the test chip in the 45-nm IBM SOI CMOS process.

3.3 Measurement Results

The chip testing was done using two different work loads to study the impact of switching activity factor: 1) ECG dataset (average switching activity = 0.065) consisting of 30-min ECG recordings of 10 patients from the MIT-BIH arrhythmia database [88], and 2) a synthetic dataset (average switching activity = 0.37). The ECG waveform was sampled at 200 Hz and quantized to 11 bits as input to the ECG processor chip. However, given a critical supply voltage $V_{dd,crit}$, the chip is operated at the corresponding critical frequency f_{crit} which is typically greater than 200 Hz. The higher frequency of operation can be used to process multiple ECG signals simultaneously. For example, modern ECG monitoring systems use 12- and 16-lead ECG sensors instead of 3 leads [81].

Figure 3.6 shows the measured energy for the conventional system (without error compensation) and the corresponding f_{crit} as a function of supply voltage. Measured results indicate that the error-free MEOP ($V_{dd,opt}$, f_{opt}, E_{min}) is (0.4 V, 600 kHz, 0.72 pJ) and (0.3 V, 65 kHz, 4.1 pJ) for the ECG and synthetic datasets, respectively. The lower $V_{dd,opt}$ for the synthetic

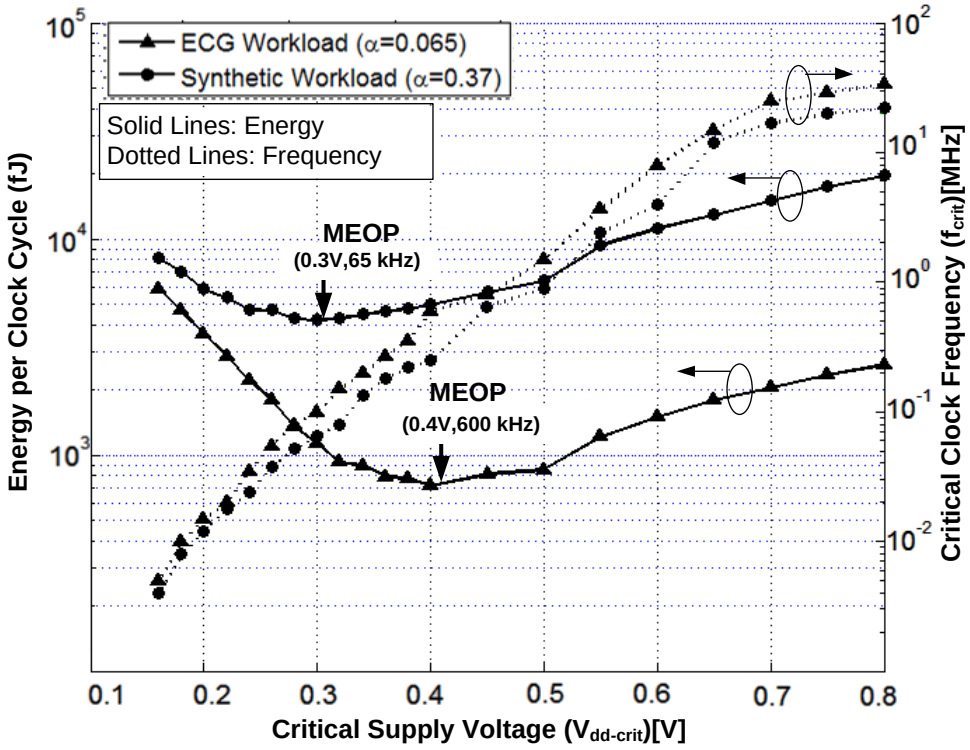


Figure 3.6: Measured energy and frequency of the conventional (error-free) ECG processor under different work loads.

dataset is expected, since a higher activity factor causes dynamic energy to be more dominant than leakage energy, and thus equilibrium is reached at a lower supply voltage.

Hardware/timing errors can be introduced by either VOS ($V_{dd} = K_{VOS}V_{dd,opt}$), FOS ($f = K_{FOS}f_{opt}$), or a combination of both to reduce energy and illustrate system robustness to hardware errors. The pre-correction error rate p_η which is the probability that the main ECG processor output (without error compensation) is in error due to VOS or FOS at MEOP is shown in Fig. 3.7 for the two datasets. The pre-correction error rate p_η increases rapidly for VOS as compared to FOS. This is expected due to the exponential dependence of delay on V_{dd} in subthreshold regime, and its linear dependence on f. Note that the synthetic dataset has higher p_η than ECG dataset, since more critical paths are being excited with higher switching activity factor.

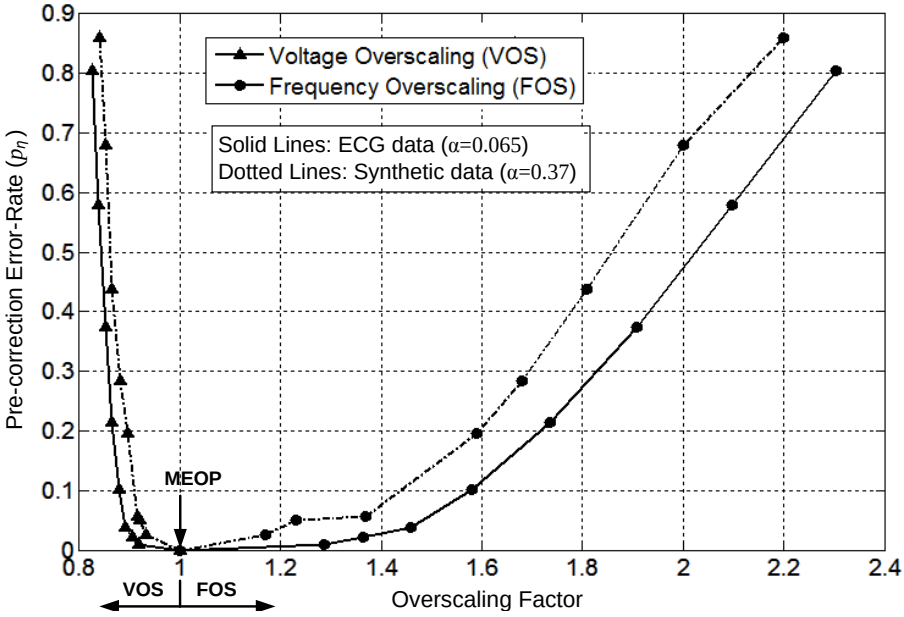


Figure 3.7: Measured pre-correction error rate at the MEOP of the ECG processor under voltage and frequency overscaling.

Employing FOS RTL-level simulations, Fig. 3.8 shows the impact of hardware/timing error rate p_η on the system-level metrics for QRS detection: 1) the sensitivity (Se) which is the probability of detecting a true QRS-complex, and 2) the positive predictivity (^+P) which is the probability that the detected QRS-complex is true [92]. These metrics are defined in terms of detection events as follows:

$$Se = \frac{TP}{TP + FN} \tag{3.1}$$

$$^+P = \frac{TP}{TP + FP} \tag{3.2}$$

where TP, FN, and FP are the number of true-positive, false-negative, and false-positive events, respectively. Probability values greater than or equal to 0.95 for Se and ^+P are desirable [92].

Two different scenarios are employed in the simulations in Fig. 3.8: 1) error-free MA (pipelining latches introduced at the outputs of the DS- and MA-block and the MA-block

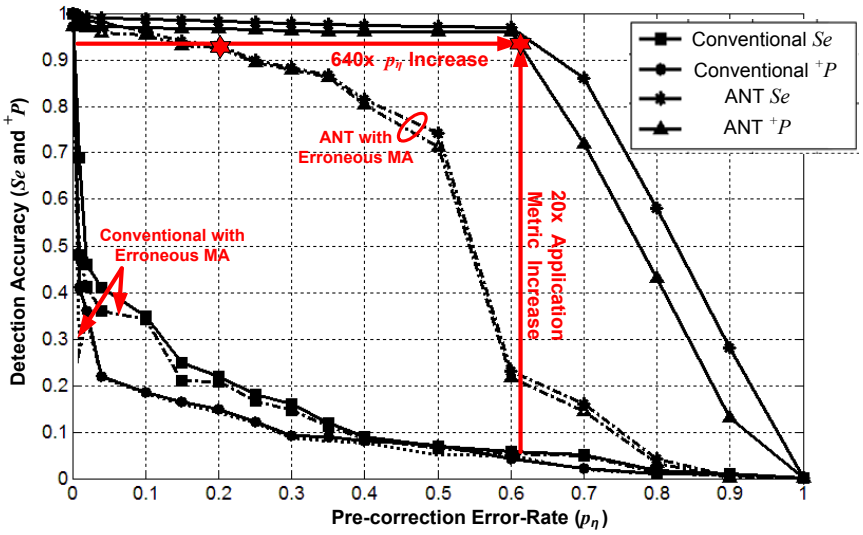


Figure 3.8: Simulated detection performance of the conventional and ANT-based ECG processors at different pre-correction error rates (solid lines indicate error-free MA and dotted lines indicate erroneous MA).

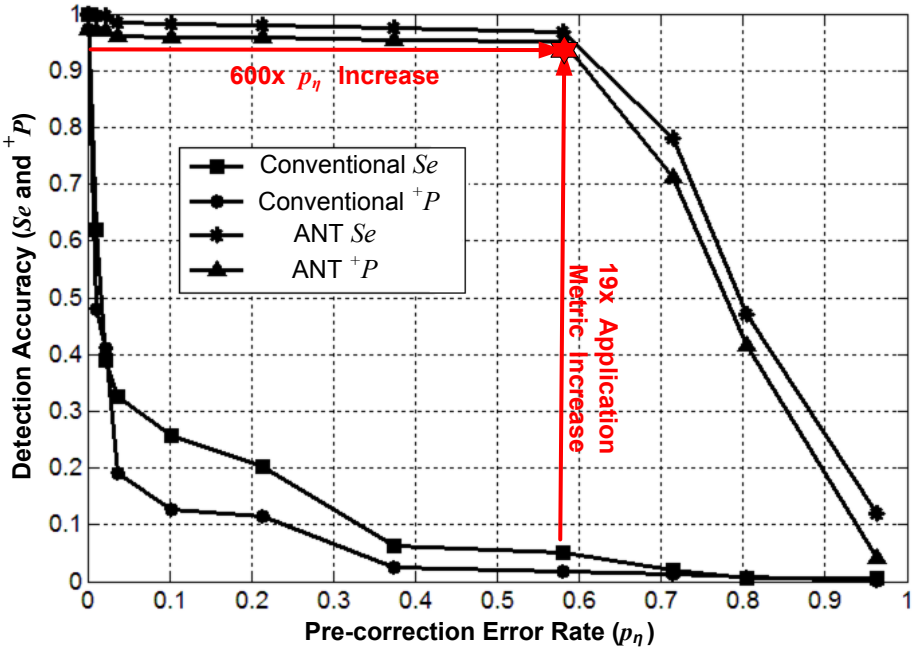


Figure 3.9: Measured detection performance of the conventional and ANT-based ECG processors at different pre-correction error rates at the MEOP while the MA block is error free.

voltage is not overscaled), and 2) erroneous MA (pipelining latch introduced at the output of the MA-block only, and all module voltages are overscaled). Error compensation is done at the output of the MA block in both cases. For case 1, $Se \geq 0.95$ and ${}^{+}P \geq 0.95$ for $p_\eta \leq 0.62$ corresponding to a 640× increase in p_η handling capability and 20× improvement in detection accuracy. For case 2, $Se \geq 0.95$ and ${}^{+}P \geq 0.95$ for $p_\eta \leq 0.2$ corresponding to a 220× increase in p_η handling capability and 4× improvement in detection accuracy. This result clearly shows the effectiveness of ANT and the intrinsic error compensating attribute of the MA-block since it acts as a low-pass filter averaging out large-magnitude errors. Note that the error-free MA-block does not help the conventional architecture which fails dramatically for $p_\eta > 0.001$. This is due to the fact that the adaptive peak-detector block in the PTA has memory, and thus uncorrected errors are propagated across different clock-cycles causing erroneous thresholds to be used for different clock-cycles. This is not the case of ANT where large errors are corrected for, at least in an approximate sense, prior to the peak-detector block.

Measurement results that demonstrates the impact of ANT are shown in Fig. 3.9 where the ECG processor with error-free MA block is voltage overscaled at its MEOP (0.4 V, 600 kHz, 0.72 pJ). The ANT-based ECG processor achieves the desired level of detection accuracy (Se and ${}^{+}P > 0.95$) in the presence of a large raw error rate $p_e \leq 0.58$, which corresponds to a 600× greater p_e handling capability, and a 19× improvement in Se and${}^{+}P$ compared to conventional error-free designs.

Comparing Figs. 3.8 and 3.9, shows that the measured results via VOS and those obtained via frequency-overscaled RTL gate-level netlist simulations match very closely at the same error-rate. In fact, Fig. 3.10 shows a close match between the measured and simulated timing error probability distribution collected at the output of the main ECG processor. Such error statistics can be further explicitly exploited by advanced statistical error compensation techniques [66] in order to further increase system robustness to hardware errors.

The instantaneous RR-interval measurement distribution for the conventional and the

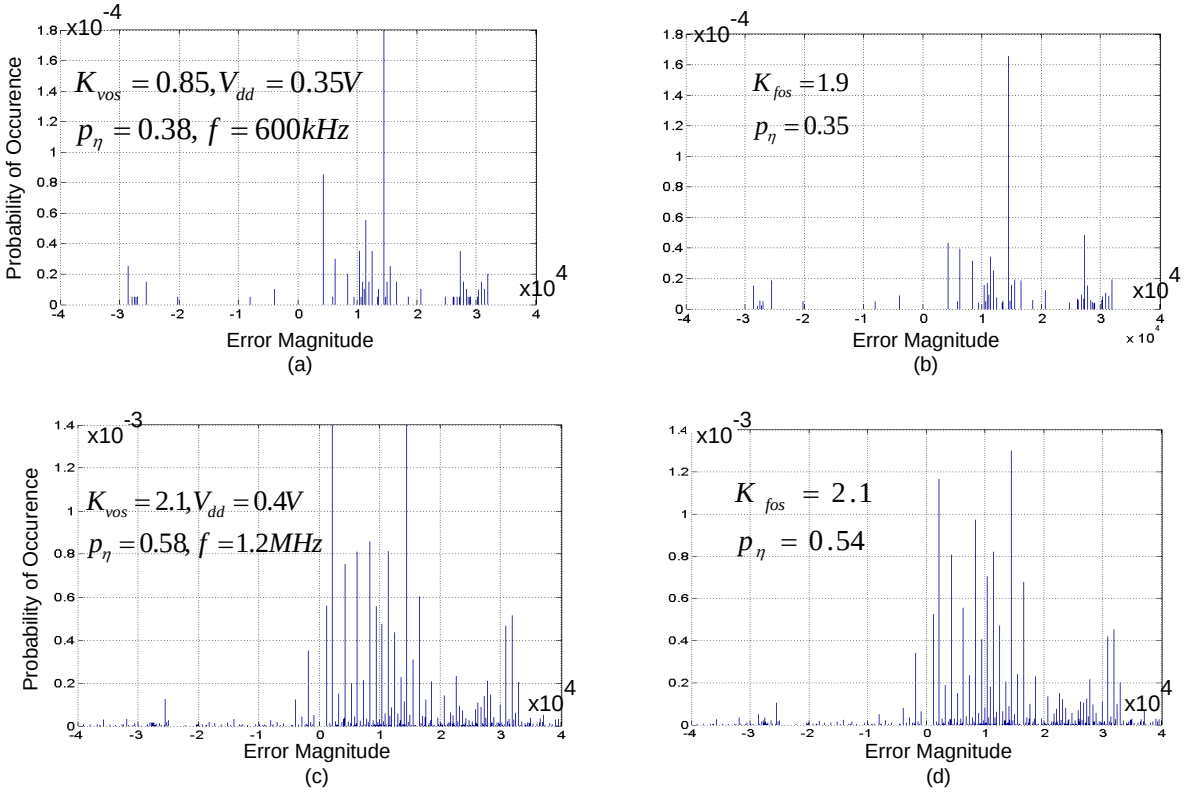
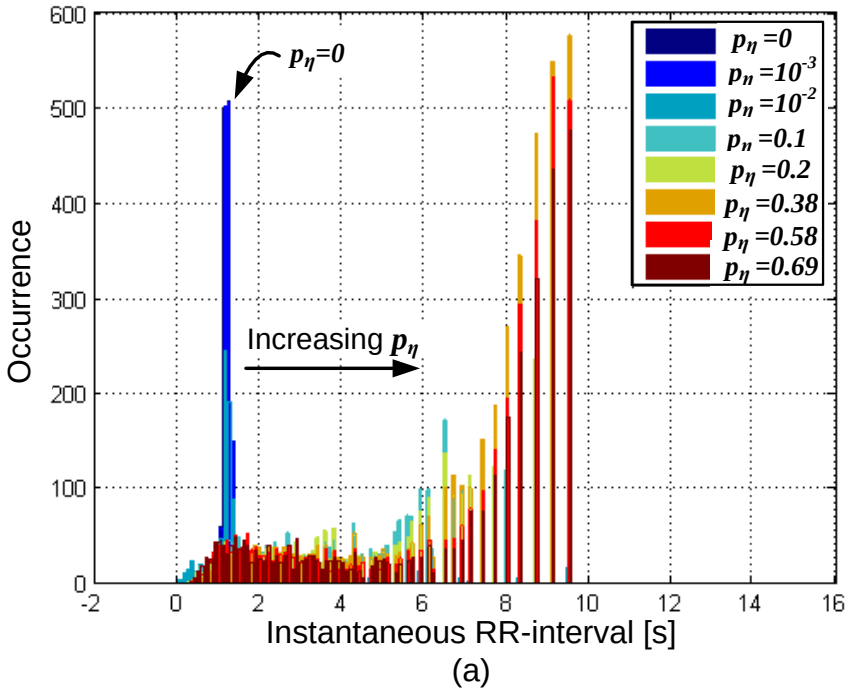


Figure 3.10: Error statistics of the ECG processor: (a) measured voltage overscaled processor with $p_\eta = 0.38$ at MEOP, (b) RTL simulations at $p_\eta = 0.35$, (c) measured frequency overscaled processor with $p_\eta = 0.58$ at MEOP, and (d) RTL simulations at $p_\eta = 0.54$.



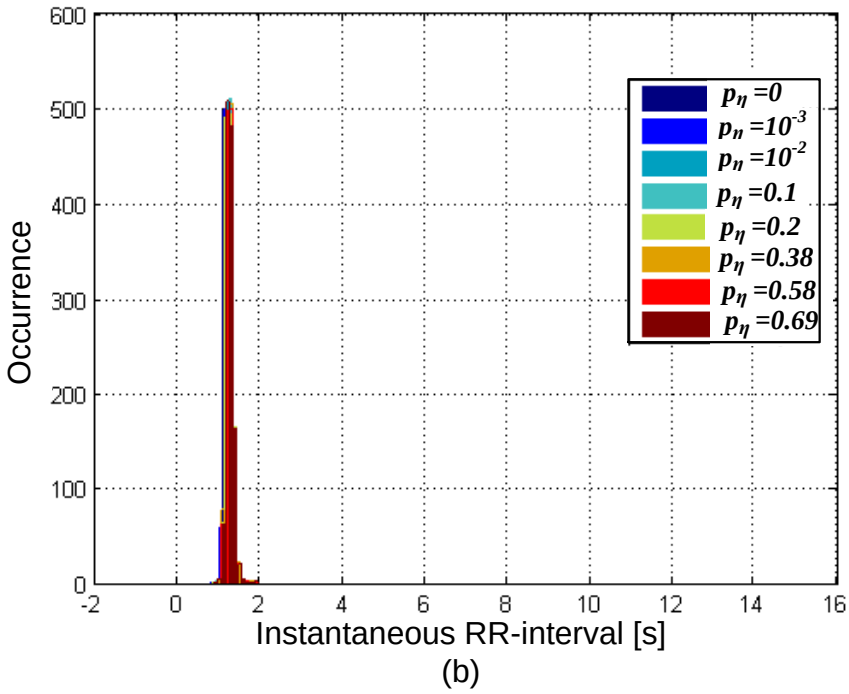


Figure 3.11: Distribution of instantaneous RR-interval measurement at MEOP: (a) conventional ECG processor and (b) ANT ECG processor.

ANT-based ECG processors is shown in Fig. 3.11 under different p_η with error-free MA. While conventional processor can maintain a reasonable RR-interval (1.2 s) only for very low p_η ($< 10^{-3}$) after which more spread is observed in RR-interval measurements, the ANT processor can maintain reasonable RR-interval up to $p_\eta = 0.58$.

Figure 3.12(a) shows measured iso-p_η contours in the V_{dd}–f plane. We refer to the ANT-based ECG processor operating on the $p_\eta = 0$ contour in Fig. 3.12(a) without error-compensation overhead as the conventional processor. Vertical translation (fixed V_{dd}) in the V_{dd}–f plane from the $p_e = 0$ contour (see Fig. 3.12(a)) corresponds to an application of FOS. Similarly, a horizontal translation (fixed f) in the V_{dd}-f plane from the $p_\eta = 0$ contour corresponds to an application of VOS. Arbitrary translations correspond to a joint application of VOS and FOS. The total energy consumption per iso-p_η contour (including the energy overhead of error compensation for $p_\eta \neq 0$) is shown in Fig. 3.12(b). The new MEOP of the ANT-based processor at $p_\eta = 0.58$ is $(0.34\,\text{V}, 630\,\text{kHz}, 0.52\,\text{pJ})$. This corresponds to a simultaneous 15% reduction in $V_{dd,opt}$, 5% increase in f_{opt}, and a 28% reduction in E_{min} compared to the MEOP of an error-free processor. Alternatively, the conventional processor with $V_{dd,crit} = 0.34\,\text{V}$ operates at $f_{crit} = 250\,\text{kHz}$ and consumes 0.9 pJ. Thus, the ANT-based processor (at its MEOP) can be viewed as being frequency overscaled with a factor of $K_{FOS} = 630/250 = 2.5$, i.e., there is 2.5× increase in throughput, along with a 42% energy savings.

Note that for $V_{dd} > 0.4\,\text{V}$ in Fig. 3.12(b), the ANT-based processor operates at higher frequency and consumes more energy than the conventional processor since FOS reduces leakage energy only, which is not sufficient to account for the energy overhead of error compensation. As V_{dd} drops below 0.4 V, leakage starts to dominate the overall energy, and hence ANT starts to show energy savings.

Similarly to Fig. 3.12, Figs. 3.13(a) and (b) shows the measured iso-p_η contours and the corresponding processor energy, respectively, for the synthetic dataset. The new MEOP of the ANT-based processor at $p_\eta = 0.58$ for the synthetic dataset is $(0.26\,\text{V}, 65\,\text{kHz}, 3\,\text{pJ})$.

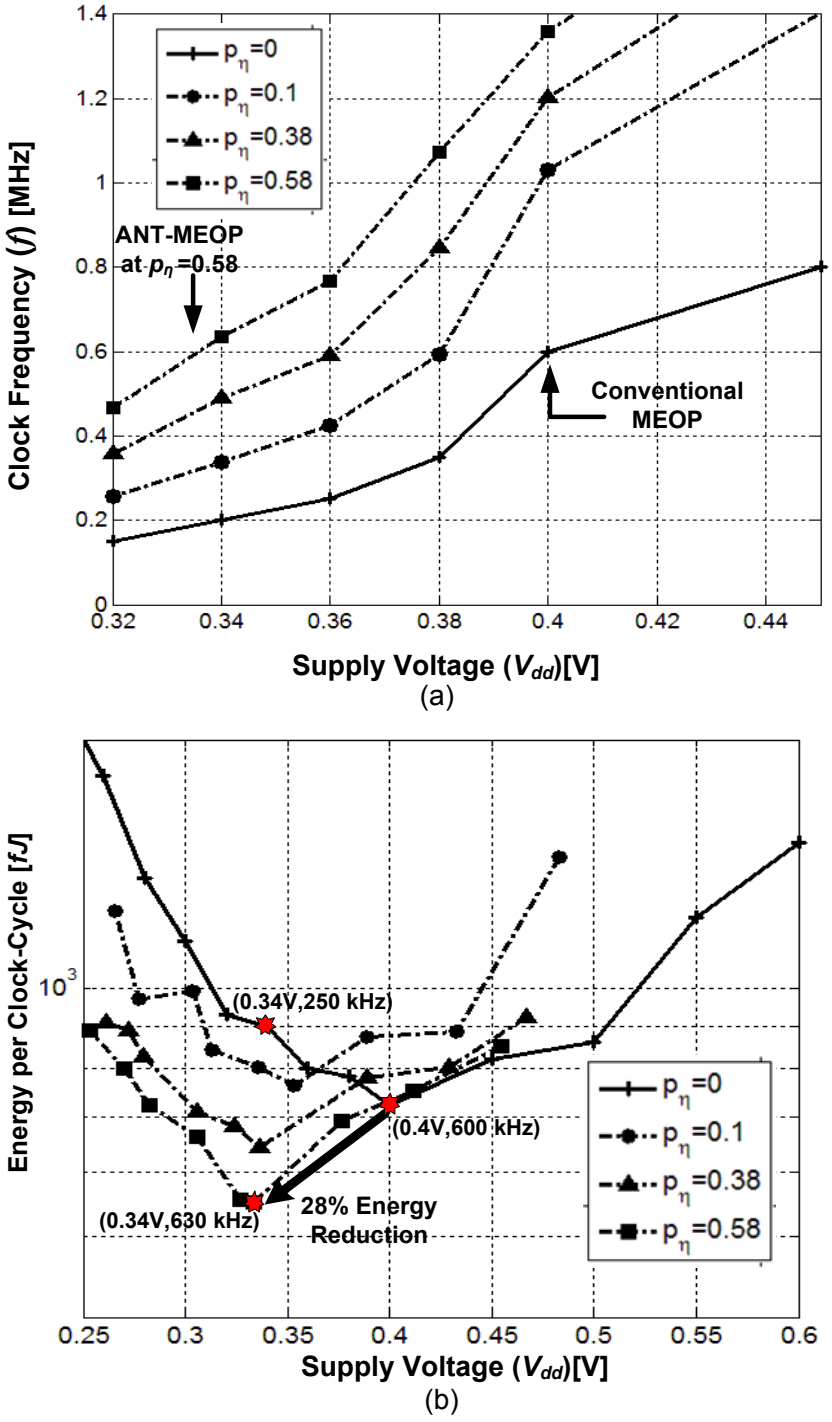


Figure 3.12: ANT-based ECG processor measurement results under the ECG dataset: (a) iso-p_η contours in the V_{dd}–f plane and (b) the total energy (including error compensation overhead for $p_\eta \neq 0$) corresponding to the iso-p_η contours.

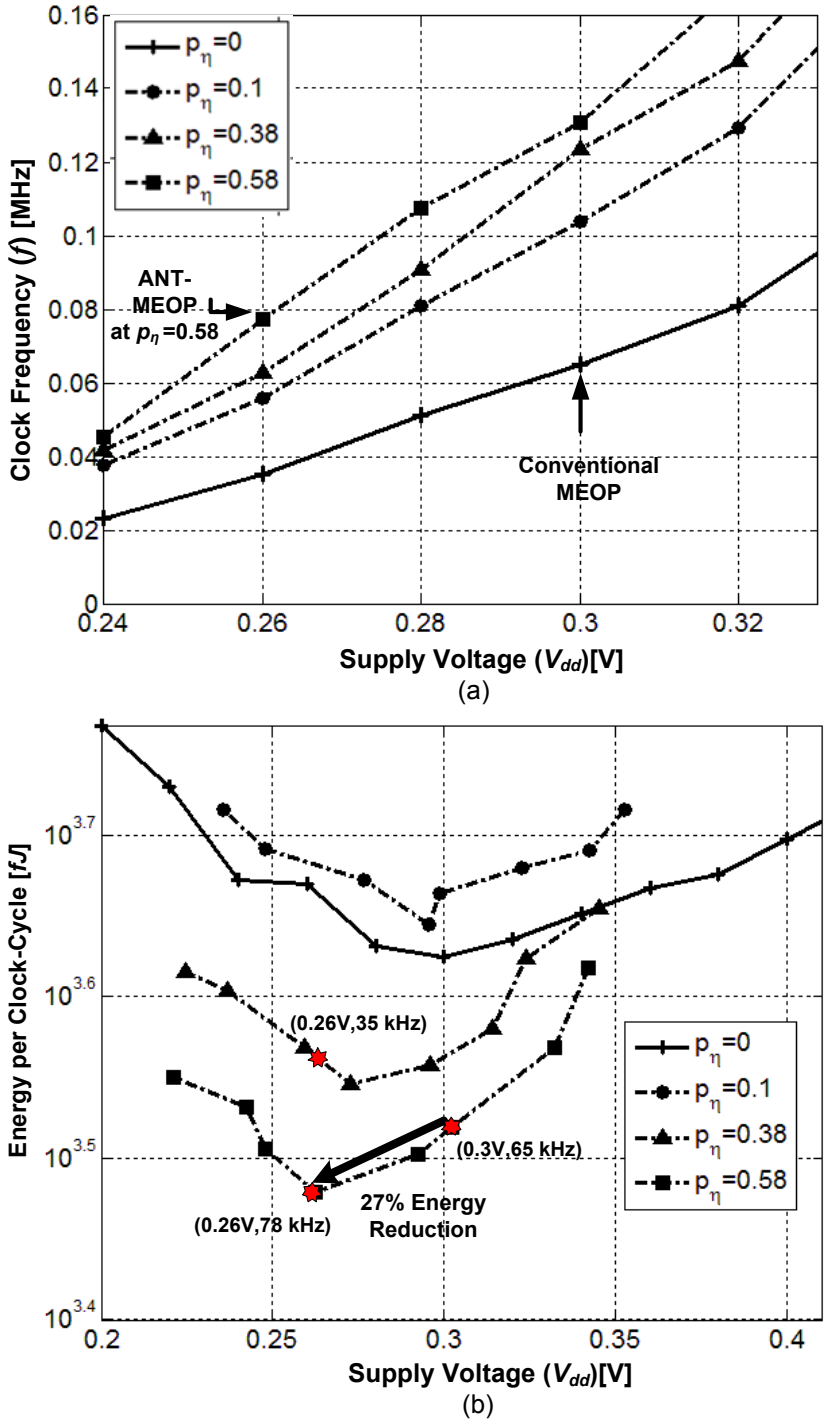


Figure 3.13: ANT-based ECG processor measurement results under the synthetic dataset: (a) iso-p_η contours in the V_{dd}–f plane and (b) the total energy (including error compensation overhead for $p_\eta \neq 0$) corresponding to the iso-p_η contours.

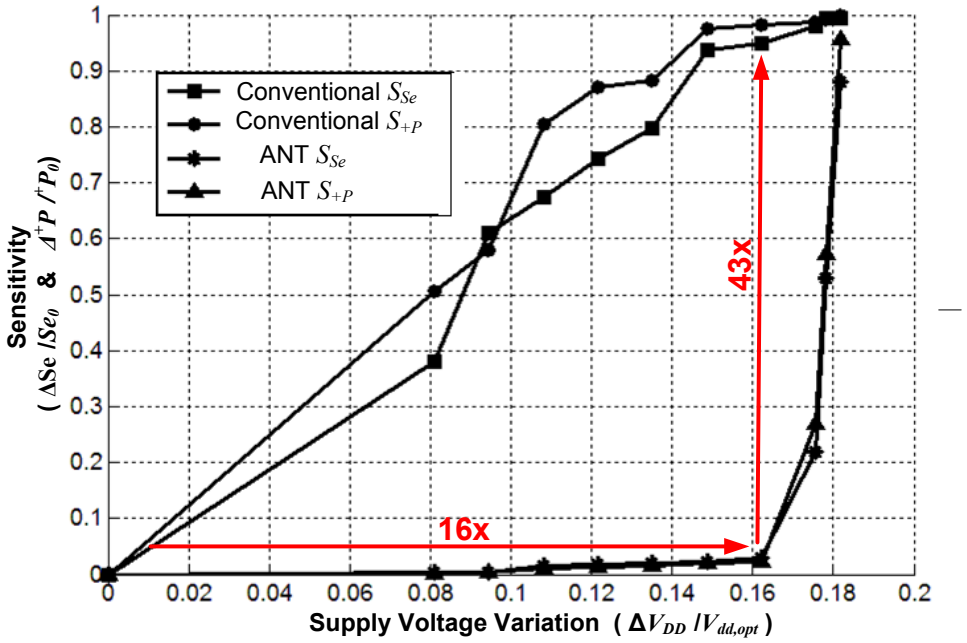


Figure 3.14: Measured sensitivity and robustness of conventional and the ANT-based ECG processors to voltage variations at the conventional MEOP (0.4 V, 600 kHz).

Table 3.2: Comparison with state-of-the-art systems.

Design Type	Near/Sub-threshold		Error Resilient			Both
	[37]	[38]	[53]	[54]	[55]	This Work
Technology (nm)	90	130	180	45	65	45
(V_{dd}[V], f[MHz])	(0.4,1)	(0.5,7)	(1.8,N.A.)	(1.165,185)	(1,3000)	(0.34,0.6)
Error Rate (p_η)	0	0	0.001	0.04	0.001	0.58
Energy/Cycle (pJ)	13	29	870	505	N.A	0.52
Energy/Cycle/ 1 k-Gates (fJ)	68	483	N.A.	8416	N.A.	15
Energy Savings (past PoFF)	0	0	14%	5%	7%	28%

This corresponds to a simultaneous 13% reduction in $V_{dd,opt}$, 20% increase in f_{opt}, and a 27% reduction in E_{min} compared to the synthetic dataset MEOP of an error-free processor. Alternatively, the conventional processor with $V_{dd,crit} = 0.26\,\text{V}$ operates at $f_{crit} = 35\,\text{kHz}$ and consumes 5 pJ. Thus, the ANT-based processor (at its synthetic dataset MEOP) can be viewed as being frequency overscaled with a factor of $K_{FOS} = 65/35 = 1.85$, i.e., there is 1.85× increase in throughput, along with a 40% energy savings.

The sensitivity of Se ($S_{Se} = \Delta Se/Se$) and ${}^{+}P$ ($S_{+P} = \Delta {}^{+}P/{}^{+}P$) to voltage variations is characterized in Fig. 3.14 at the conventional MEOP supply voltage $V_{dd,opt} = 0.4\,\text{V}$, where we find that the ANT-based processor tolerates up to 16× higher voltage variations, and shows up to 43× lower sensitivity (S_{Se} and S_{+P}) compared to the conventional processor.

Table 3.2 compares our design to other near or subthreshold $p_e = 0$ biomedical processors. The ANT-based ECG processor consumes 14.5 fJ/cycle/1k-*gate* which is at least 4.7× more energy efficient than state-of-the-art in addition to the robustness benefits. As comparing to other error-resilient designs in superthreshold, the ANT-based ECG processor tolerates an error rate of up to $p_e < 0.58$, which is at least 580× greater than existing techniques. Therefore, we see that stochastic computating provides tremendous increase in robustness while meeting the threshold of acceptable detection performance. In addition, it results in up to a 28% reduction in energy beyond the minimum achievable energy.

3.4 Summary

This chapter presented a stochastic computing-based ECG processer in a 45-nm IBM CMOS process. The prototype IC illustrates the robustness and energy benefits of stochastic computing in the subthreshold regime, where robustness is more of a concern due to increased sensitivity to voltage, process, and temperature variations. The IC shows robust operation up to a 58% error rate along with a 28% energy reduction beyond minimum achievable energy and is 4.7× more energy efficient than state-of-the-art.

CHAPTER 4

JOINT OPTIMIZATION OF POWER DELIVERY AND CORE ENERGY IN ULP PLATFORMS

This chapter addresses the problem of designing energy-efficient embedded systems by jointly optimizing the energy of both the DC-DC converter(s) and the computational core(s) in ULP platforms. Chapters 2 and 3 demonstrated the existence of a minimum energy operating point (MEOP) in the subthreshold region for conventional and stochastic cores (core(C)-MEOP) defined by the energy-optimum core voltage $V_{dd,opt}$, the energy-optimum frequency f_{opt} and the minimum core energy consumption E_{min}. In energy-aware ULP platforms, the core supply voltage V_C or V_{dd} (see Fig. 4.1(b)) is generated by a programmable DC-DC converter, which translates a higher battery voltage to a lower core voltage $V_{dd} < 1V$ and dynamically adjusts it depending on the workload characteristics. Past work has focused primarily on independently optimizing the energy consumption of the core and the DC-DC converter, especially at MEOP and in the presence of DVS, leading to sub-optimal solutions.

This chapter proposes to minimize system (core and DC-DC converter) energy consumption by jointly optimizing the DC-DC converter and core. Architectural-level techniques (see Fig. 4.1(a)), which exploits joint-design principles, application-level requirements, and stochastic core robustness, are proposed to mitigate energy-delivery losses. First, we show dynamic voltage scaling (DVS) causes the overall system MEOP (S-MEOP) to differ significantly from C-MEOP due to the increased DC-DC converter losses. Simulations in a 130-nm, 1.2 V IBM CMOS process show that operation at S-MEOP results in a 45.5% energy savings over operation at a core voltage $V_{C,opt}$ suggested by C-MEOP. The DC-DC converter efficiency is also improved by 2.2×. Second, we show that architectural techniques cause the S-MEOP to approach C-MEOP. Thus, it is sufficient to track C-MEOP – a much easier task

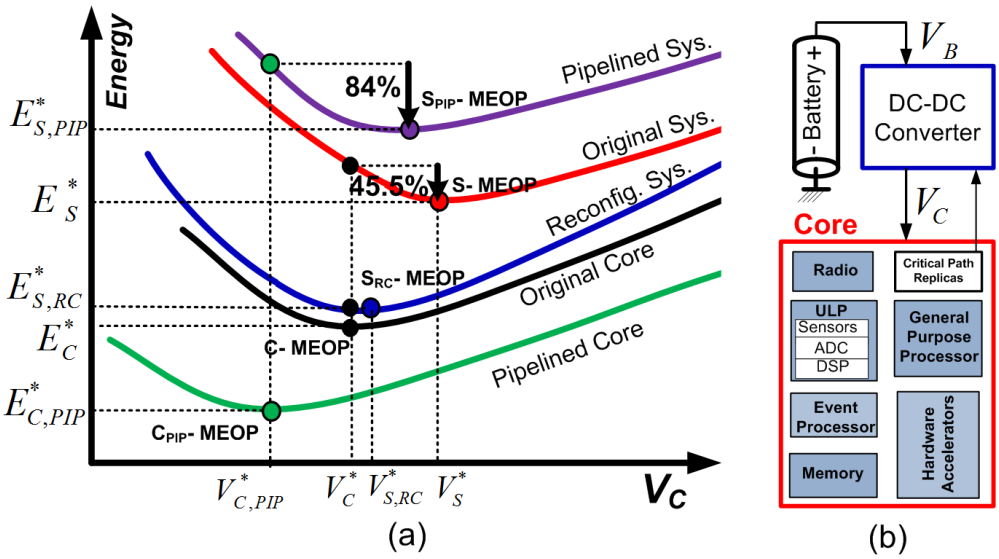


Figure 4.1: Energy-aware embedded system: (a) energy under DVS and (b) block diagram.

on-chip – in order to account for process variations and changing work load characteristics. We show that core parallelization reduces DC-DC converter losses in the subthreshold regime but increases it in the superthreshold regime. This observation leads us to propose a reconfigurable core architecture that improves the converter efficiency by 2.3× at C-MEOP, and makes energy consumption at S-MEOP and C-MEOP to be within 4% of each other. This also improves throughput in the subthreshold regime by at least 8×. Furthermore, we show that pipelining, which has been proposed to decrease core energy at C-MEOP while improving throughput [28], adversely affects the S-MEOP. Pipelined system's energy at S-MEOP is 85% lower than when operating at the C-MEOP voltage $V_{C,opt}$. The DC-DC converter efficiency is also improved by 10-percentage points compared to the unpipelined-core system. Finally, we address the energy delivery for stochastic compute cores. The robustness of a stochastic core to voltage variations relaxes the voltage ripple specification, and thus reducing DC-DC losses and improving system energy efficiency. Preliminary results demonstrate the promise of joint stochastic compute core and DC-DC converter design, and open up interesting further investigations and future extensions.

This chapter is organized as follows: Section 4.1 analyzes the core energy consumption

in DVS and the associated DC-DC losses. Section 4.3 discusses the challenges in designing efficient DC-DC for a wide range of load conditions, and illustrates the energy gains obtained by operating at S-MEOP instead of C-MEOP through simulations in 130-nm IBM CMOS process. Section 4.4 presents the proposed reconfigurable core-architecture and joint-system design techniques to reduce energy at S-MEOP and improve DC-DC efficiency.

4.1 Core Energy Characterization Under Dynamic Voltage Scaling (DVS)

The core energy varies widely depending on the workload and application throughput requirements. Dynamic voltage scaling (DVS) employs a programmable DC-DC converter to adjust the core supply voltage V_C, to meet the application throughput requirements at minimum energy consumption.

The two main sources of core energy are *dynamic* and *leakage* energy (E_{dyn} and E_{lkg}) and the latter becomes significant only when the core operates in subthreshold regime ($V_C \leq V_{th}$ where V_{th} is the threshold voltage). The core energy E_C, can be expressed as:

$$\begin{aligned} E_C &= E_{dyn} + E_{lkg} \\ E_{dyn} &= \alpha N C_g V_C^2 \\ E_{lkg} &= \frac{N I_{OFF} V_C}{f_C} \end{aligned} \tag{4.1}$$

where α is the average switching activity factor of the core gates, N is the number of core gates each with an output load capacitance C_g, f_C is the core operating frequency, and I_{OFF} is the OFF-state leakage current.

The MOSFET drain current, I_D as a function of gate-to-source and drain-to-source volt-

ages (V_{GS}, V_{DS}) in the subthreshold and superthreshold regimes, is given by:

$$I_D\left(V_{GS}, V_{DS}\right) = \begin{cases} I_o e^{\frac{V_{GS}-V_{th}-\gamma V_{DS}}{mV_T}}\left(1 - e^{\frac{-V_{DS}}{V_T}}\right) \\ \quad \text{if } V_{GS} < V_{th} + \nu m V_T \\ I_o e^{\frac{\nu m V_T + \gamma V_{DS}}{mV_T}}\left(\frac{V_{GS}-V_{th}}{\nu m V_T}\right)^{\nu} \\ \quad \text{if } V_{GS} \geq V_{th} + \nu m V_T \end{cases} \tag{4.2}$$

where I_o is a reference current and is proportional to the transistor W/L ratio, ν is the velocity saturation index, m is the subthreshold slope factor, γ is the DIBL coefficient, V_{TH} is the threshold voltage, and V_T is the thermal voltage. Using (4.2), the ON-state and OFF-state currents for an NMOS transistor are $I_{ON} = I_D\left(V_C, V_C\right)$ and $I_{OFF} = I_D\left(0, V_C\right)$, respectively.

Assuming the critical path of the core has a logic depth of K gates each with an output load capacitance C_g, the core operating frequency f_C is given by:

$$f_C = \frac{I_{ON}}{\beta K C_g V_C} \tag{4.3}$$

where β is a fitting parameter needed to match the finite signal rise and fall times. The subthreshold frequency decreases exponentially with V_C reduction in subthreshold due to the exponential dependance of I_{ON} on V_C in (4.2) when $V_{GS} < V_{th} + \eta m V_T$. This leads to an exponential increase in subthreshold *leakage* energy. Leakage energy as a function of V_C is obtained by substituting (4.3) in (4.1) to yield:

$$E_{lkg} = \beta N K C_g V_C^2 \frac{I_{OFF}}{I_{ON}} = \begin{cases} \beta N K C_g V_C^2 e^{\frac{-\gamma V_C}{mV_T}} \\ \quad \text{if } V_C < V_{th} + \nu m V_T \\ \beta N K C_g V_C^2 h \frac{1-e^{\frac{-V_C}{V_T}}}{(V_C - V_{th})^{\nu}} \\ \quad \text{if } V_C \geq V_{th} + \nu m V_T \end{cases} \tag{4.4}$$

where h is a constant function of V_{th}, V_T, ν, and m. Note in superthreshold, leakage is negligible and varies as $1/(V_C - V_{th})^{\nu}$ since $V_C >> V_T$. Including the dynamic energy, the total core energy is given by:

$$E_C = \begin{cases} NC_gV_C^2\left(\alpha + \beta K e^{\frac{-\gamma V_C}{mV_T}}\right) \\ \quad \text{if } V_C < V_{th} + \nu m V_T \\ NC_gV_C^2\left(\alpha + \beta K h \frac{1-e^{\frac{-V_C}{V_T}}}{(V_C - V_{th})^{\nu}}\right) \\ \quad \text{if } V_C \geq V_{th} + \nu m V_T \end{cases} \tag{4.5}$$

Therefore, decreasing V_C results in a quadratic reduction in E_{dyn} at the expense of increased delay or reduced frequency of operation f_C. As V_C is reduced below V_{th}, i.e., subthreshold operation, E_{lkg} increases very rapidly and becomes comparable to E_{dyn}. This trade-off between E_{dyn} and E_{lkg} in subthreshold is well studied [23] [25], and results in a minimum energy operating point (MEOP) defined via the tuple (E_C^*, V_C^*, f_C^*).

4.2 Design and Analysis of DC-DC Converters

The programmable DC-DC converter efficiency greatly depends on the core energy E_C. The DC-DC converter regulates the core supply voltage V_C (see Fig. 4.1(b)) from an external battery with voltage V_B, which is greater than V_C. It is imperative to decrease the losses of the DC-DC converter to maximize the efficiency of energy delivery. Three key types of DC-DC converters are linear regulator (LR), switched-capacitor regulator (SC), and switching regulator (SR). LR employs a power MOSFET whose gate is controlled by a feedback error-control signal to supply a specific current demand from the battery while maintaining a constant V_C. The efficiency of LR is limited by the ratio of V_C/V_B. Add to it the losses in the control driver and power MOSFET. SC delivers the energy to the core by discharging the battery energy through a capacitive network exchange. SC achieves efficiency better

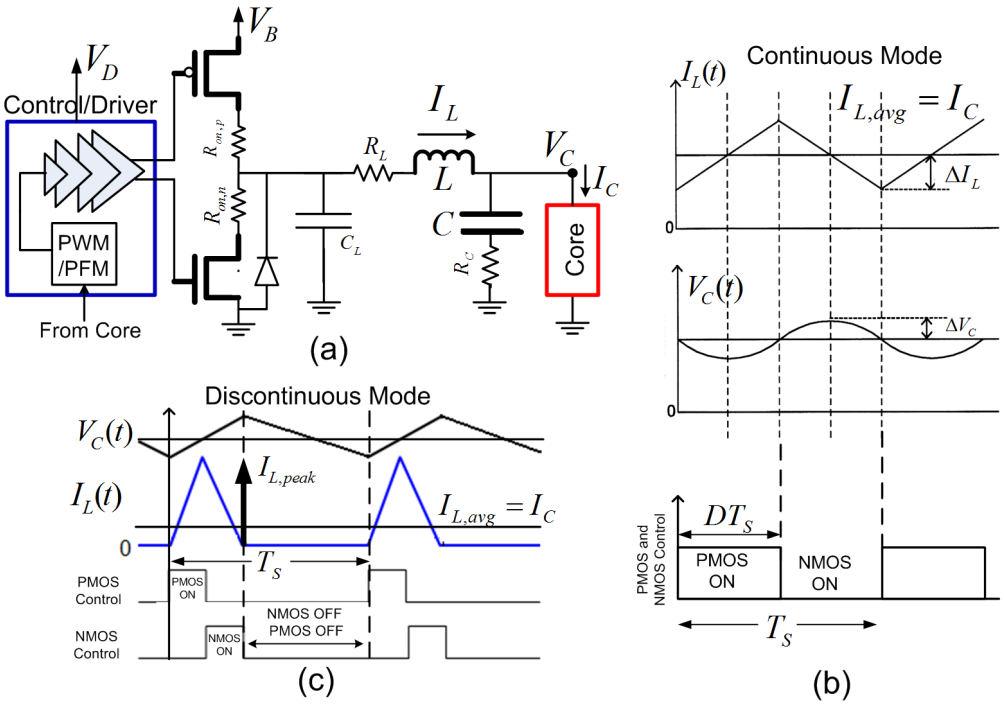


Figure 4.2: Switching DC-DC converter: (a) block diagram with parasitics, (b) continuous-conduction mode (CCM), and (c) discontinuous-conduction mode (DCM).

than LR but has poor output regulation due to output voltage ripple. In addition, SC does not allow continuous DVS because the ratio of voltage conversion(s) $D = V_C/V_B$ is already determined by the chosen capacitor values and topology. The SRs are most widely used because they enable continuous DVS with high efficiency over a relatively wide range of load variations, and they are well studied in the literature [93] [94]. The block diagram of an SR DC-DC converter is shown in Fig. 4.2(a). Commonly, pulse-width modulation (PWM) is employed to generate a periodic pulse with duty cycle D and switching frequency $f_s = 1/T_s$. The periodic pulse controls the gates of a PMOS and an NMOS switch. When the pulse is low, the NMOS is off and the PMOS is turned on, and current is supplied from the battery to the load (continuous-conduction mode (CCM)) (see Fig. 4.2(b)). At light (low-current) loads, the DC-DC enters discontinuous-conduction mode (DCM) (a variable PWM or pulse-frequency modulation (PFM)) to decrease switching losses and improve efficiency. In DCM,

there are durations when both NMOS and PMOS are turned off when the inductor current (I_L) reaches zero in order to prevent it from flowing in the reverse direction (see Fig. 4.2(c)).

The LC-filter acts as a low-pass filter to pass the average of $I_L(t)$ as the core current $I_C(t)$ and blocks the AC component of core voltage V_C (see Fig. 4.2(b) and (c)). Typically, the duty cycle D is chosen to be equal V_C/V_B so that $I_{L,avg} = I_C$. The output core voltage ripple is given by:

$$\frac{\triangle V_C}{V_C} = \frac{1-D}{16LCf_s^2} \tag{4.6}$$

The choices of LC-filter (passive) components and the converter switching frequency f_s, determine the output voltage ripple and are chosen to balance switching and conduction losses. Increasing f_s decreases the size of the passive components and the conduction losses associated with the LC-parasitics, but increases the switching losses.

The performance and losses of switching DC-DC converter are well studied in the literature [94] [93]. The losses mainly include the conduction losses, switching losses, and drive losses. The conduction losses (P_{cond}) are due to the ON-resistance of PMOS and NMOS switches ($R_{on,p}$ and $R_{on,n}$, respectively), the inductor parasitic resistance (R_L), and the capacitor effective series resistance R_C (see Fig. 4.2(a)) and are related to the root-mean square currents through the inductor, PMOS, and NMOS switch in CCM and DCM. The conduction loss through a resistance R with varying current and voltage terminals is given by $I_{R,rms}^2 R$ where $I_{R,rms}$ is the resistor root-mean-square (RMS) current. The RMS currents through the resistances in Fig. 4.2(a) are largely determined by the inductor current waveform $I_L(t)$ whose average value yields the supplied core current I_C (see Fig. 4.2(b) in CCM and (c) DCM). In CCM, the RMS currents through $R_{on,p}$ and $R_{on,n}$ are, respectively:

$$\begin{aligned} I_{rms,p} &= \sqrt{D\left(I_C^2 + \frac{\triangle i_L^2}{3}\right)} \\ I_{rms,n} &= \sqrt{(1-D)\left(I_C^2 + \frac{\triangle i_L^2}{3}\right)} \end{aligned} \tag{4.7}$$

where $D = V_C/V_B$ and $\triangle i_L$ is the ripple in $I_L(t)$ in Fig. 4.2(b) and is given by:

$$\triangle i_L = \frac{V_C(1-D)}{2Lf_{sw}} \tag{4.8}$$

In DCM, the RMS currents through $R_{on,p}$ and $R_{on,n}$ are:

$$\begin{aligned} I_{rms,p} &= \sqrt{\frac{Lf_{sw}I_{L,peak}}{3(V_B - V_C)}} \\ I_{rms,n} &= \sqrt{\frac{Lf_{sw}I_{L,peak}}{3V_C}} \end{aligned} \tag{4.9}$$

where $I_{L,peak}$ is the peak inductor current in Fig. 4.2(c) and is given by:

$$I_{L,peak} = \sqrt{\frac{2I_C V_C(1-D)}{Lf_{sw}}} \tag{4.10}$$

The switching losses (P_s) are due to the current and voltage overlap when activating the PMOS and NMOS switches and are given by $P_s = \frac{1}{a}\tau V_B I_C$ where I_C is core current, a is a number usually between 2 and 6 that describes the switching trajectory, and τ is the percentage of the DC-DC switching period when the switch current and voltage overlap. The drive losses (P_{drive}) are due to the capacitive switching in the MOSFET switch driver and the controller and are given by $P_{drive} = f_s C_d V_d^2$, where C_d is the driver and controller switching capacitance and V_d is the driver supply voltage. The DC-DC converter efficiency (η_{DC}) and energy (E_{DC}) are given by:

$$\begin{aligned} \eta_{DC} = \frac{P_C}{P_C + P_{DC}} &= \frac{P_C}{P_C + P_{cond} + P_s + P_{drive}} \\ E_{DC} &= \frac{P_{DC} f_C}{f_s} \end{aligned} \tag{4.11}$$

where P_{DC} is the power losses in the DC-DC converter, P_C is the core power, and E_{DC} is the DC-DC energy loss per core instruction.

4.3 System (Core and DC-DC Converter) Energy Optimization

We optimize the total system (core and DC-DC converter) energy consumption in presence of variations in core energy demand due to DVS employing the energy models in Sections 4.1 and 4.2 and HSPICE simulations in a 1.2 V 130-nm IBM CMOS process.

We model the computational core as a bank of 50 16-b×16-b multiply-accumulate (MAC) units. Each MAC unit (see Fig. 4.3(a)) operates at a core voltage and frequency (V_C, f_C) and computes $y[n] = y[n-1] + x_1[n] \times x_2[n]$ where $x_1[n]$ and $x_2[n]$ are 16-b input signals, $y[n]$ is a 32-b output, and n is the clock-cycle/time index. We employ a ripple carry-based architecture with 1-b full adders as the basic building block to study the energy consumption of the core. Figures 4.3(b) and (c) show the core frequency and energy $(50 \times E_{MAC})$ based on the analytical models in (4.3) and (4.5), respectively, and HSPICE simulations of the circuit schematic of the MAC unit at various core voltages V_C for workloads with average switching activity of $\alpha = 0.3$ and 0.1. The analytical models in (4.3) and (4.5) approximate the results of HSPICE simulations very well, and hence will be employed to estimate core energy in the rest of the chapter. As voltage is reduced to subthreshold, f_C decreases exponentially and E_{lkg} increases significantly while E_{dyn} continues to decrease. Figure 4.3(c) shows that the C-MEOP is reached at $(E_C^* = 60\,\text{pJ}, V_C^* = 0.33\,\text{V}, f_C^* = 1.5\,\text{MHz})$ for a workload with $\alpha = 0.3$. We can see that as V_C varies from 1.2 V down to $V_C^* = 0.33\,\text{V}$, the frequency f_C and energy consumption E_C vary by 200× and 9×, respectively, or a 1800× variation in power demand. Figure 4.3(c) shows that the average switching factor impacts the dynamic energy only. Thus, in the rest of this paper, we focus on energy demand variations due to DVS only and consider a workload with a fixed average switching factor of $\alpha = 0.3$.

We assume a 3.3 V external battery source, and the DC-DC converter designed at a switching frequency of $f_s = 10\,\text{MHz}$ while maintaining an output ripple of around 10% for all V_C. This leads to relatively reasonable passive element values of $L = 94\,\text{n}H$ and $C = 47\,\text{n}F$, resulting in reduced conduction losses and improved efficiency (> 80%) in the

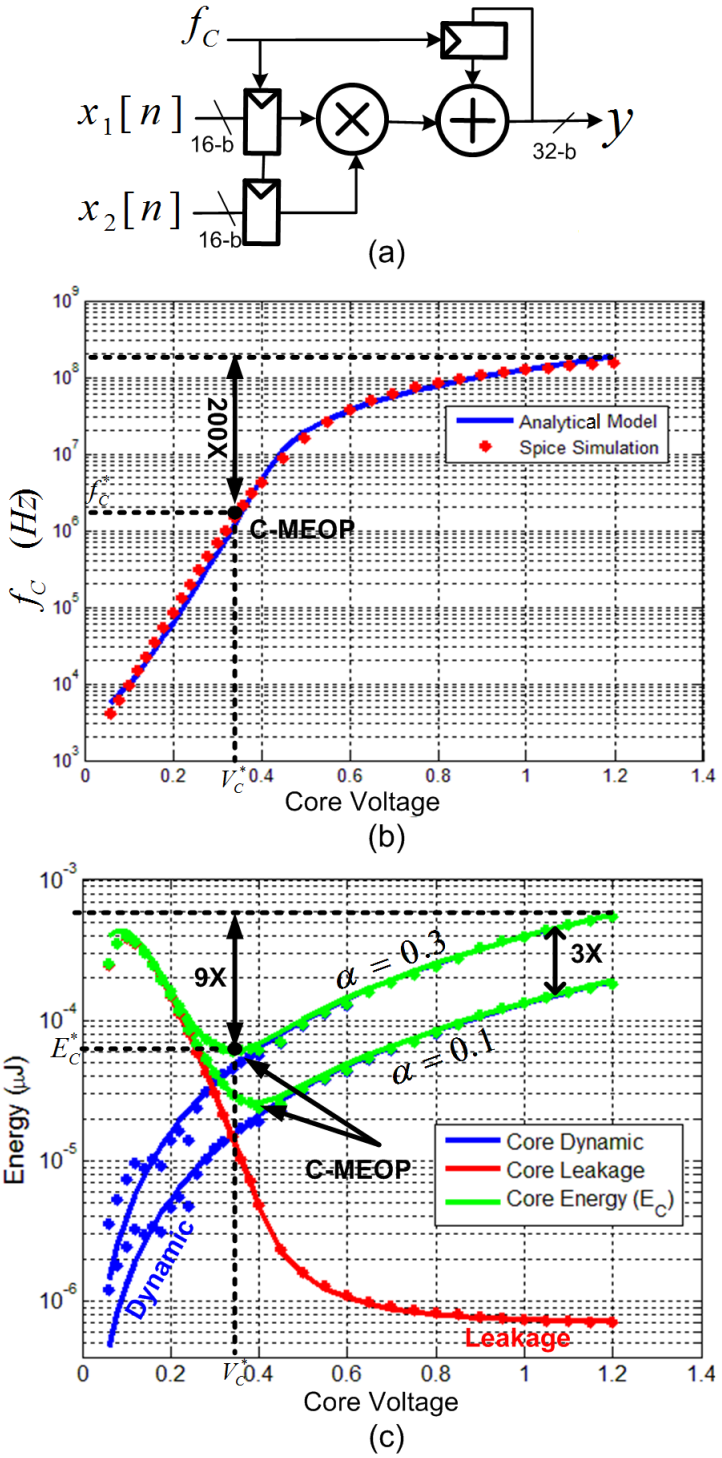


Figure 4.3: The computing core model: (a) architecture of a single MAC unit, (b) the core frequency, and (c) the core energy consumption under DVS.

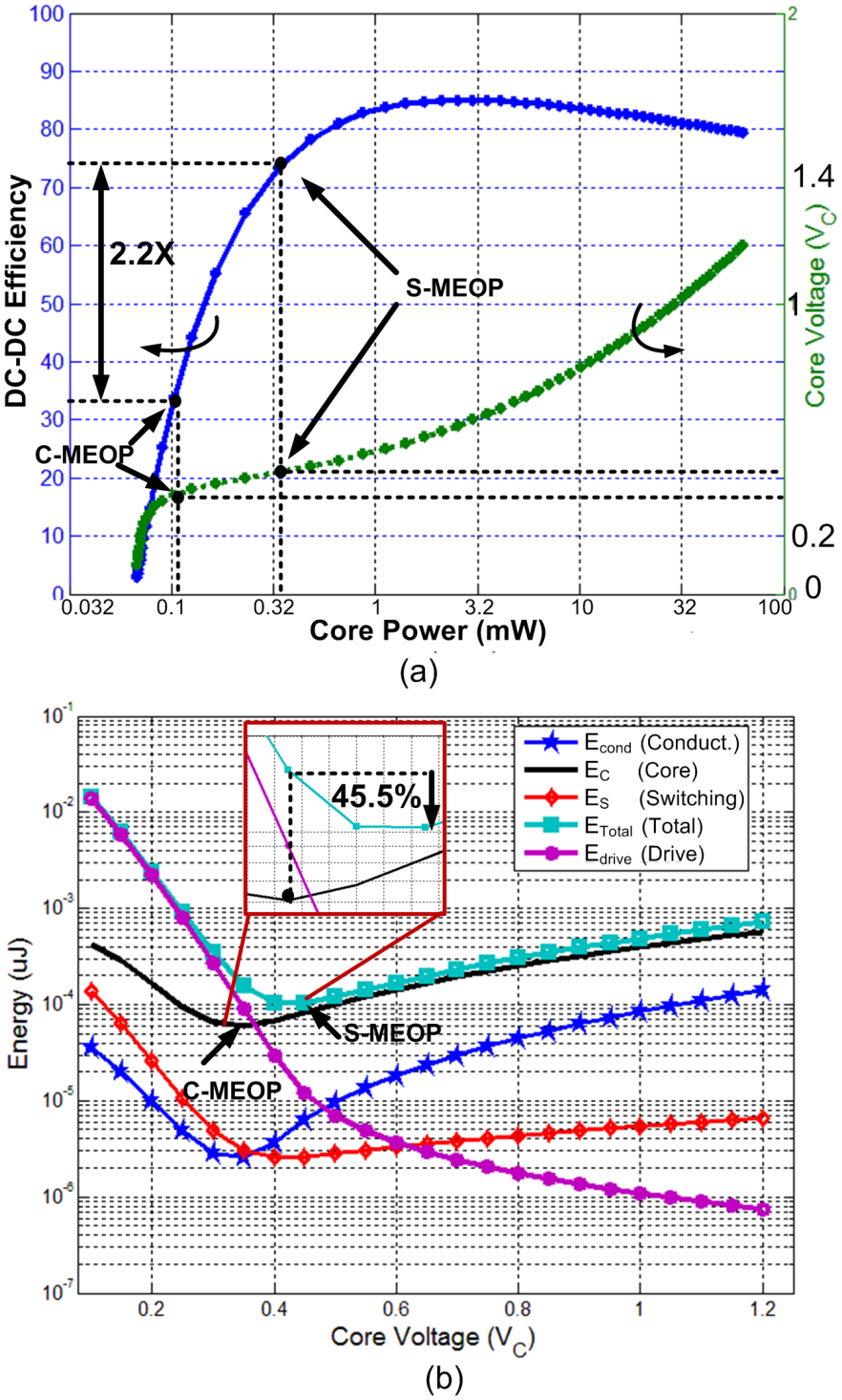


Figure 4.4: DVS system energy:(a) DC-DC efficiency vs. core power and supply voltage and (b) the total system energy and losses.

superthreshold regime. In fact, the converter can maintain an efficiency greater than 80% for $0.45\,\mathrm{V} \leq V_C \leq 1.2\,\mathrm{V}$ while delivering a core power in the range 0.6 to 50 mW (see Fig. 4.4(a)). When V_C is decreased further, efficiency drops significantly reaching 33% at the C-MEOP determined previously in Fig. 4.3(c). This is due to the drive energy per core instruction $E_{drive} = P_{drive}/f_C$ loss dominating in the subthreshold regime. The energy losses of the DC-DC converter per core clock-cycle is shown in Fig. 4.4(b). Here, it is assumed that the capacitance of the driver C_d is equivalent to 1% of the core capacitance NC_g, and its voltage $V_D = 1.2\,\mathrm{V}$. While conduction energy per core instruction $E_{cond} = P_{cond}/f_C$ and switching energy $E_s = P_s/f_C$ losses scale well with the core energy E_C under DVS, the drive energy E_{drive} losses increase significantly in subthreshold since f_C starts to decrease exponentially as compared to the DC-DC switching frequency f_s. The converter switching frequency f_s does not decrease much with V_C in subthreshold under DCM because the output ripple needs to be maintained at less than 10%. The increased driver losses cause the total system MEOP (S-MEOP) core voltage V_S^* to be higher than that V_C^* at the C-MEOP. Operating at V_S^* instead of V_C^* results in 2.2× improvement in efficiency and 45.5% energy savings as illustrated in Fig. 4.4(a) and (b), respectively. However, tracking S-MEOP on-chip is more difficult than C-MEOP as it requires external feedback. Next, we study core architecture techniques to aid the design of energy efficient DC-DC converters at the C-MEOP in DVS and thus increase the system efficiency and make S-MEOP approach C-MEOP.

4.4 Core Architecture Optimization for Energy-Efficient Systems

To minimize DC-DC converter drive losses per core instruction E_{drive} in subthreshold, we employ architecture techniques to increase the core operating frequency f_C so that the controller in discontinuous mode can better adapt its DC-DC switching frequency (f_s) to the core frequency while maintaining less than 10% output ripple core voltage.

4.4.1 Energy-Efficient Multicore Systems

Parallelization/unfolding are commonly used techniques to increase core throughput. Parallelizing/unfolding by a factor of M, will instantiate M copies of the core running at the same frequency as the original core. This increases both the throughput and power by a factor of M, and maintains the energy per instruction the same as a single core (SC) if the overhead of serialization/deserialization is ignored, i.e., the parallelized core (PC) or multicore operates at the same energy level of an SC (SC-MEOP is same as PC-MEOP) while delivering higher throughput. However, parallelization increases the DC-DC converter conduction power losses P_{cond} by a factor greater than M due to the reduced conduction efficiency when delivering $M\times$ the original load power. This translates to lower system efficiency and higher system energy consumption when conduction losses dominate the DC-DC converter losses. On the other hand, the switching and driver power losses in the DC-DC converter are relatively independent of the core parallelization leading to a $M\times$ decrease in their energy overhead per instruction, since PC throughput increases by M.

The effect of parallelization on the DC-DC efficiency is shown in Fig. 4.5. Core parallelization will reduce the efficiency significantly in superthreshold, where conduction losses dominate. On the other hand, it helps to extend the DC-DC converter high-efficiency range into the subthreshold regime by reducing driver and switching losses until the core frequency decreases significantly again. This improves DC-DC efficiency at MEOP by a factor of at least $2.2\times$ for $M = 4$, and that increases for higher values of M at the expense of reduced efficiency in superthreshold (see Fig. 4.5). This motivates the use of reconfigurable core (RC) architecture under DVS. The RC uses a single core while the rest of $M-1$ cores being power-gated as long as $f_C \geq 0.1f_s$, i.e., the DC-DC controller can adapt its losses with the core frequency and maintain good output ripple. When $f_C < 0.1f_s$, all M cores are activated, which gives an additional window/slack to reduce V_C while keeping driver losses within bounds.

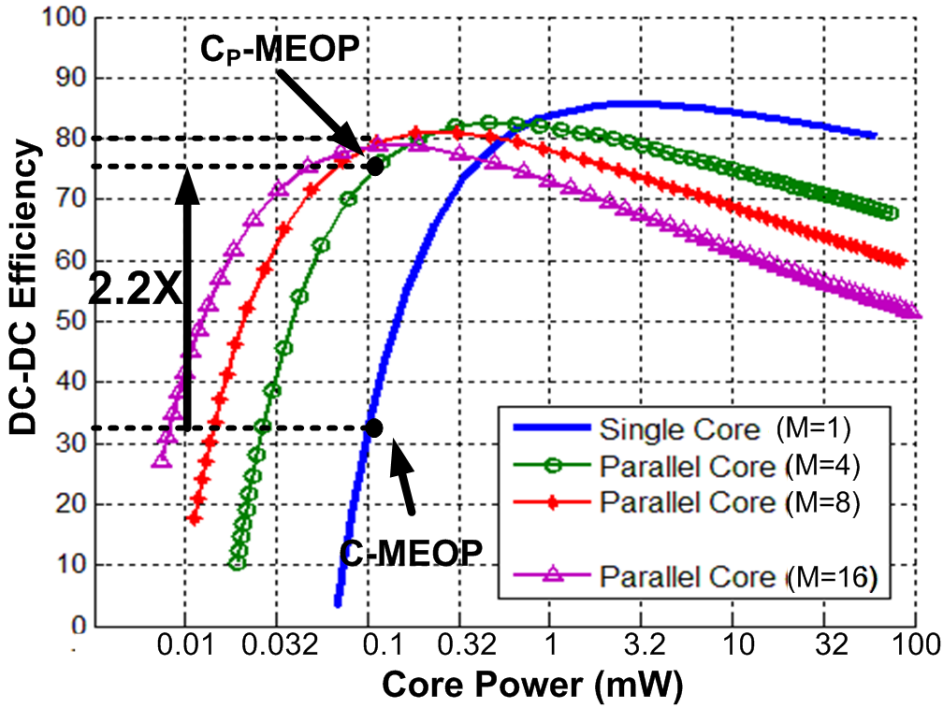


Figure 4.5: DC-DC efficiency for parallel/multi-cores.

The DC-DC efficiency of RC for $M = 8$ is shown in Fig. 4.6(a) and its corresponding energy consumption and losses are shown in Fig. 4.6(b). RC reduces the driver energy losses around C-MEOP so that S_{RC}-MEOP approaches that of C-MEOP. The total system energy at C-MEOP is within 4% of the energy at S_{RC}-MEOP, and the proposed RC efficiency at C-MEOP is 2.6× better than that of a SC efficiency. Simulations show that the difference between S-MEOP and C-MEOP under RC architecture decreases further for higher values of M, and higher efficiency is achieved. An additional benefit of the RC system architecture is that it allows higher throughput ($M = 8$×-increase) to be met in subthreshold region compared to SC system.

4.4.2 Energy-Efficient Pipelined Systems

Recently [28], pipelining has been shown to be an attractive technique in the subthreshold region as it reduces the energy consumption at C-MEOP by 30% and simultaneously

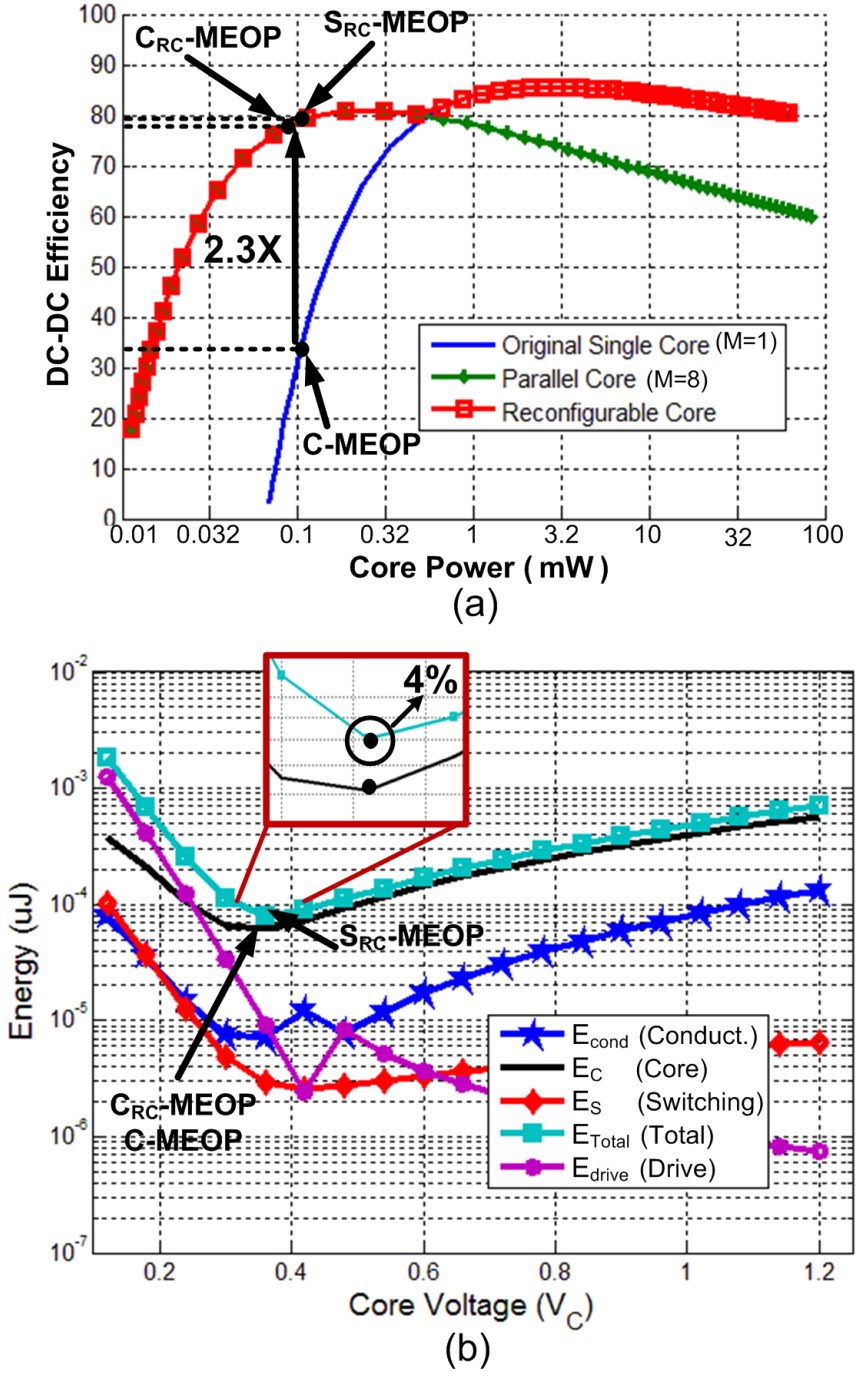


Figure 4.6: Architecture-optimized DVS system energy: (a) DC-DC efficiency and (b) reconfigurable core (RC) system energy profile.

increases throughput by 1.6×. In this subsection, we show that pipelining is unattractive when DC-DC converter losses are included.

Pipelining by a level of J will decrease DC-DC driver losses per core instruction E_{drive} by a factor of J assuming it operates at $J\times$ higher operating frequency. However, doing so increases the load current leading to an increase in conduction losses by a factor greater than J, which is why in Fig. 4.7(a), the efficiency of the pipelined system is always less than that of the original core. Figure 4.7(b) shows the energy consumption of the original core with pipelining level of $J = 4$. Pipelining reduces core leakage energy due to increased operating frequency with minimal effect on core dynamic energy (overhead of pipelining register). This reduces not only core energy at C_{PIP}-MEOP but also pushes the voltage at C_{PIP}-MEOP to lower values (compare C_{PIP}-MEOP in Fig. 4.7(b) to C-MEOP in Fig. 4.4(b)). Thus, DC-DC driver losses E_{drive} will be more significant when included in overall system energy. This leads to an 85% increase in the pipelined-core system energy if operating at C_{PIP}-MEOP instead of S_{PIP}-MEOP, in addition to a 2.6× reduction in DC-DC converter efficiency η_{DC}. Simulations also show that similar results are obtained with increased pipelining levels until pipelining-register overhead starts to dominate.

4.4.3 Energy Delivery for Stochastic Compute Cores

Chapters 2 and 3 demonstrated the robustness of stochastic-computing cores to voltage variation through their ability to tolerate a large number of voltage-induced timing errors. Such resiliency can be exploited in a joint stochastic system (DC-DC converter and stochastic-computing core) design (see Fig. 4.8) by relaxing the core voltage ripple specification. This reduces the value of the required converter passive elements (L and C) or its switching frequency f_s as shown in (4.6), and thus can reduce the form-factor or the energy conversion sub-system. Although the ripple voltage is a dynamic phenomena and is a strong function of the core work load characteristics, a conservative approach is adopted here, voltage ripple

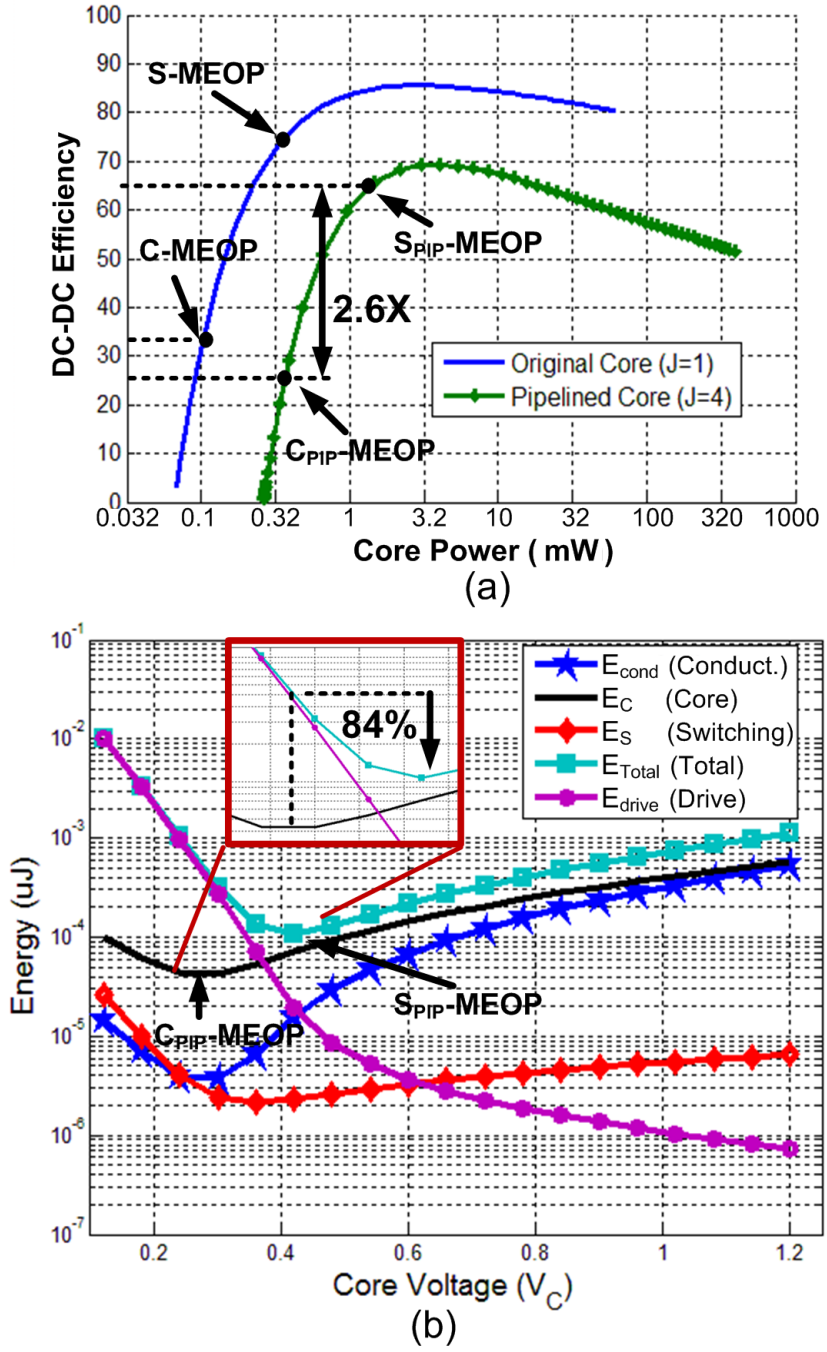


Figure 4.7: DVS system energy with core pipelining: (a) DC-DC efficiency, and (b) pipelined-core system energy profile.

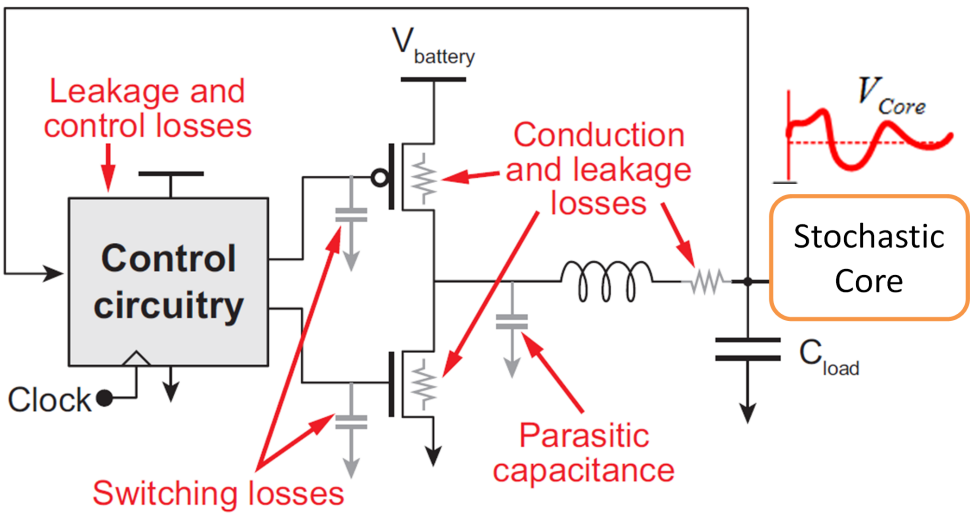


Figure 4.8: Block diagram of a stochastic system (stochastic core and DC-DC converter).

and droop is assumed to be similar to (static) VOS. Thus, since Chapters 2 and 3 showed that the stochastic core at MEOP can tolerate up to 15% worst-case voltage reduction, it is assumed that the core ripple specification can be relaxed by an additional 15%. Furthermore, since further work is needed to study the core energy and timing errors under voltage ripple and droop, we assume a worst-case scenario where the energy of the relaxed-ripple stochastic core is same as that of a conventional core.

Using the same value of L and C used so far, the DC-DC converter frequency f_s is decreased until (4.6) is satisfied with the relaxed ripple specification. Figure 4.9 shows that the switching, conduction, and driver losses are reduced once the ripple specification is relaxed (compare the dotted lines to the solid lines in Fig. 4.9). This results in 13.5% total system energy reduction at the new stochastic-system (SS)-MEOP as compared to conventional S-MEOP. In addition, the core voltage at SS-MEOP is brought closer than the core voltage at S-MEOP to the core voltage at C-MEOP. The reduction in DC-DC losses improves the DC-DC efficiency at SS-MEOP by 8 percentage points (compare SS-MEOP and S-MEOP in Fig. 4.10). All this illustrates the promise of joint optimization of DC-DC converter and stochastic core in improving overall system energy efficiency, although assuming a worst-case ripple voltage scenario. This motivates the investigation of stochastic

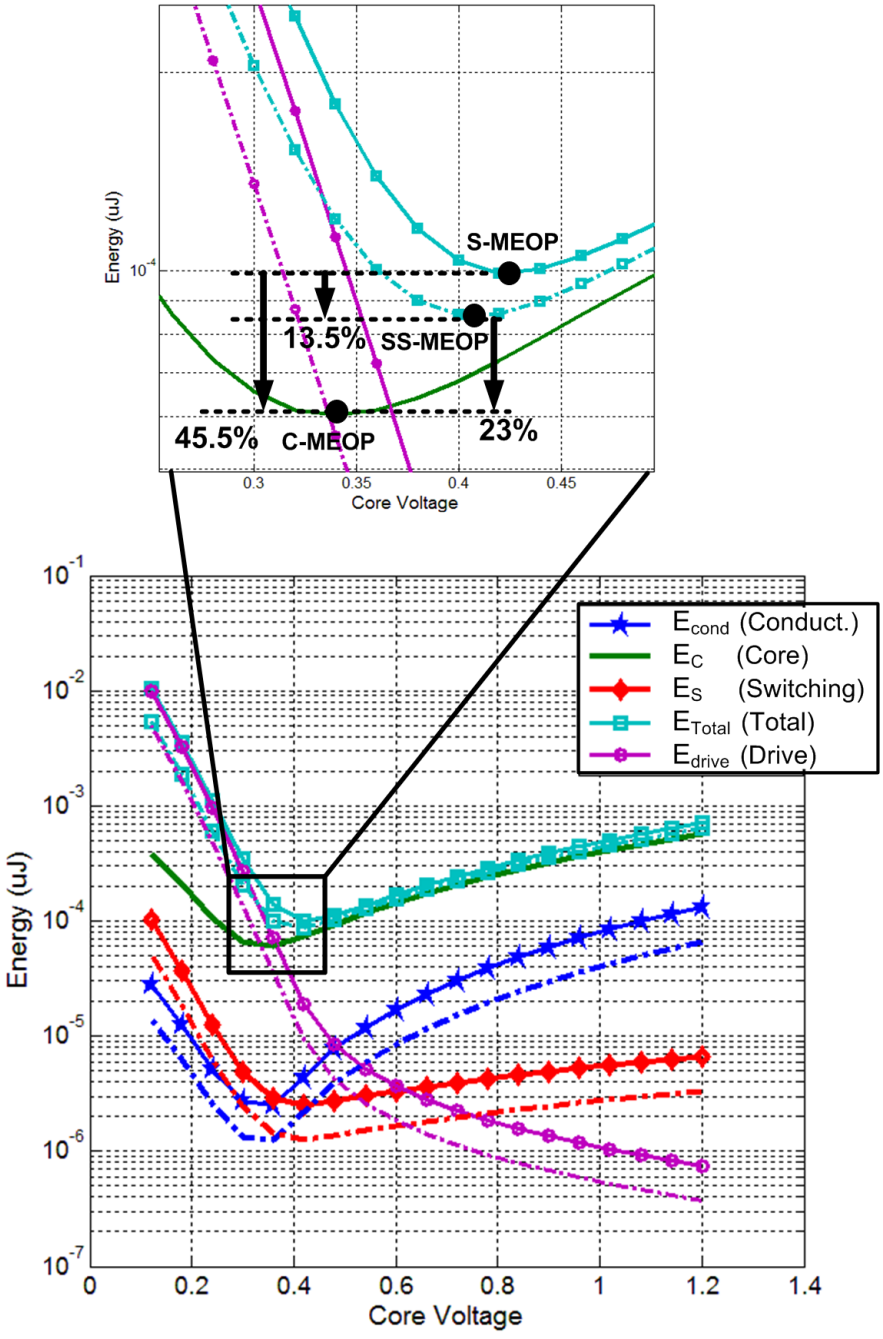


Figure 4.9: DVS energy of jointly optimized systems. Solid lines (dotted lines) refer to the conventional (stochastic) system.

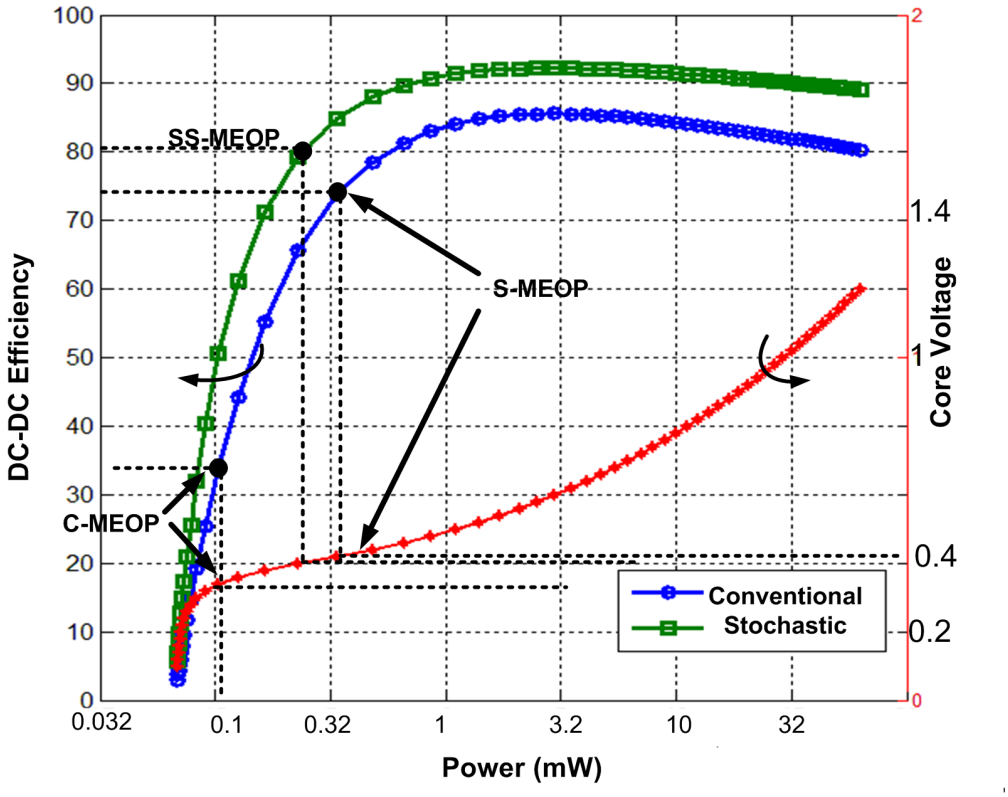


Figure 4.10: DC-DC energy efficiency of jointly optimized stochastic core and DC-DC converter.

cores under voltage-ripple induced errors and a better understanding of core-voltage ripple effect on the overall system behavior and energy which is part of the future work opened up by this dissertation.

4.5 Summary

We developed a holistic view to energy-efficient system design, taking into consideration the DC-DC converter losses. We showed that the DC-DC losses have considerable effect on minimum energy operation. We employed architecture techniques to alleviate the DC-DC converter losses so that high energy efficiency can be maintained over a wide range of core energy demand variations. Furthermore, the robustness to voltage variations of

stochastic cores is exploited to increase the DC-DC design margins and improve overall system efficiency.

CHAPTER 5

STOCHASTIC COMPUTING PLATFORMS VIA LIKELIHOOD PROCESSING

Previous chapters employed ANT in subthreshold designs. This chapter proposes a new stochastic computing technique referred to as *likelihood processing* (LP) and demonstrates its energy savings and robustness benefits in 45-nm CMOS.

Stochastic computing techniques such as ANT [70], and SSNOC [74] *implicitly* employ error statistics of the architectural components, while soft NMR [78] does so explicitly. The proposed technique of LP explicitly employs error statistics for error compensation. It does so by computing the likelihood, i.e., the ratio of the probability of an output bit being a '1' vs. the probability of it being a '0'. Doing so, LP offers a better processing of error statistics than soft NMR which exploits error statistics only at word-level. Furthermore, unlike soft NMR, LP avoids the need for replication. Results show that LP significantly improves on the robustness and energy efficiency of the conventional (error-free), ANT, and soft NMR systems. This is demonstrated in the design of a 2D discrete-cosine transform (2D-DCT) codec, a widely used image processing kernel in a TI 45-nm CMOS process. The codec can be employed as a hardware accelerator in an ULP platform for surveillance applications, for example. The energy-robustness trade-off in LP is evaluated in the presence of *voltage overscaling* (VOS) are employed to emulate timing violations due to PVT variations as well.

The chapter is organized as follows: Section 5.1 presents a unified framework for error-resiliency and stochastic-computing techniques including LP. Section 5.2 formally describes the algorithmic basis for LP, its architecture, and presents a motivational example. Section 5.3 demonstrates the benefits of LP in terms of robustness and energy efficiency in the design of a 2D-DCT image codec.

5.1 A Unified Framework for Error Resiliency

This section presents a unified framework for describing error-resiliency techniques, including both conventional and statistical techniques, in order to relate the proposed LP technique to existing work.

The error model for an arbitrary computational kernel M (see Fig. 5.1(a)) is given by:

$$y = y_o + \eta + \epsilon = y_o + e \tag{5.1}$$

where y is a B_y-bit observed output, y_o is the correct (error-free) output, η and ϵ are the hardware and estimation errors, respectively, and $e = \eta + \epsilon$ is the composite error. Though y_o can belong to a set of acceptable outputs, we take a conservative approach and assume that the cardinality of such a set is unity, i.e., there is one ideal/error-free value for y_o. The set of all possible outputs of M is referred to as the *output space* $\mathcal{Y}$, i.e., y_i, y_o, and $e \in \mathcal{Y}$. Note that, without any loss of generality, we employ a weighted number representation, e.g., two's complement, for the output and error signals, which is quite appropriate for media kernels, as these tend to employ arithmetic operations quite extensively. This does not preclude the use of other number representations both weighted and non-weighted.

In this chapter, the hardware error η arises due to timing violations, and it is typically large in magnitude because the arithmetic operations in DSP kernels are least-significant bit (LSB) first. This is reflected in the sample probability mass function (PMF) $P_\eta(\eta)$ in Fig. 5.1(b). On the other hand, the estimation error ϵ is small in magnitude (see $P_\epsilon(\epsilon)$ in Fig. 5.1(b)) because it can arise as a difference between y_o and an internal signal in block M, or perhaps the output of another lower-complexity block M_E that tries to approximate/estimate the output of M, i.e., an estimator. The topic of error characterization of a computational block is an interesting one in its own right, and can be accomplished in many ways, both off-line as well as via *in situ* calibration using typical inputs. Characterization of timings error for

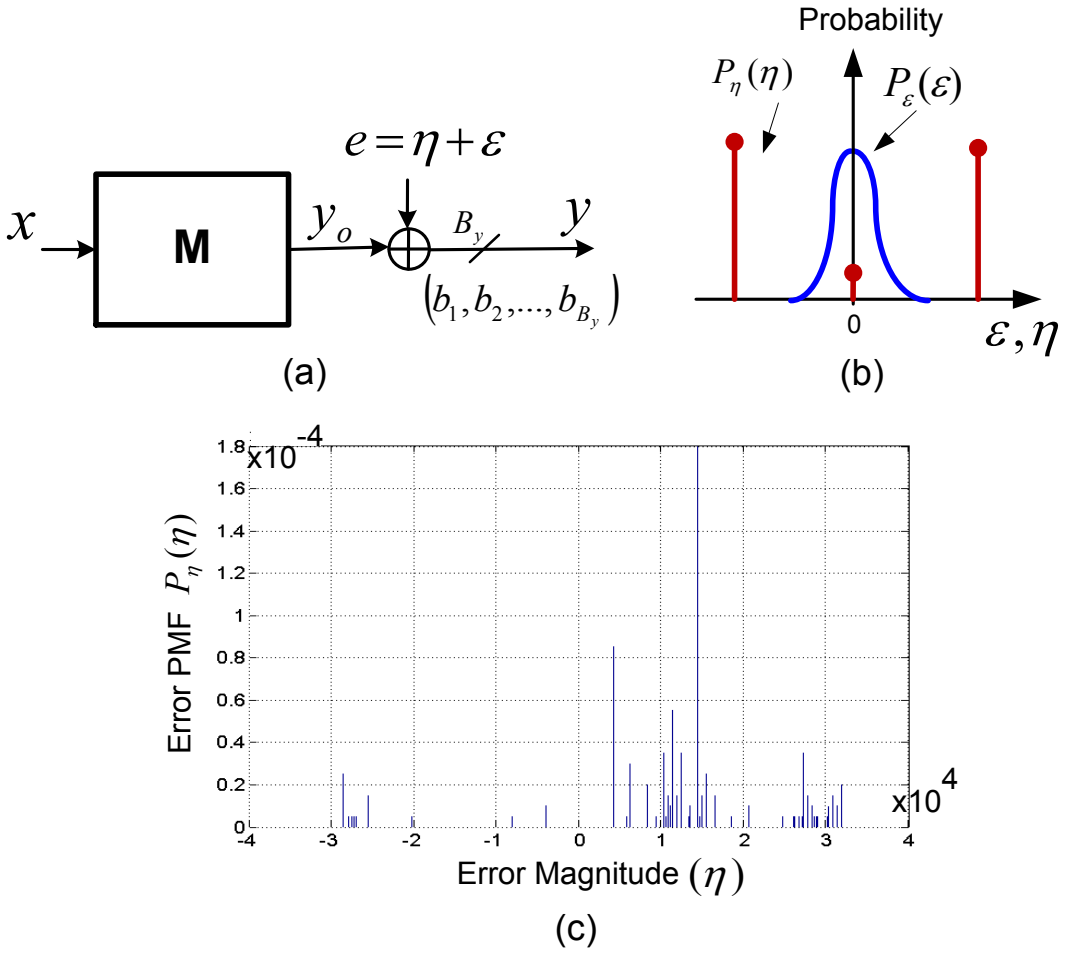


Figure 5.1: Computational error model: (a) additive error model, (b) sample error statistics, and (c) measured error PMF $P_\eta(\eta)$ of a 20-bit output filter IC in 45-nm CMOS with $V_{dd} = 0.85V_{dd-crit}$.

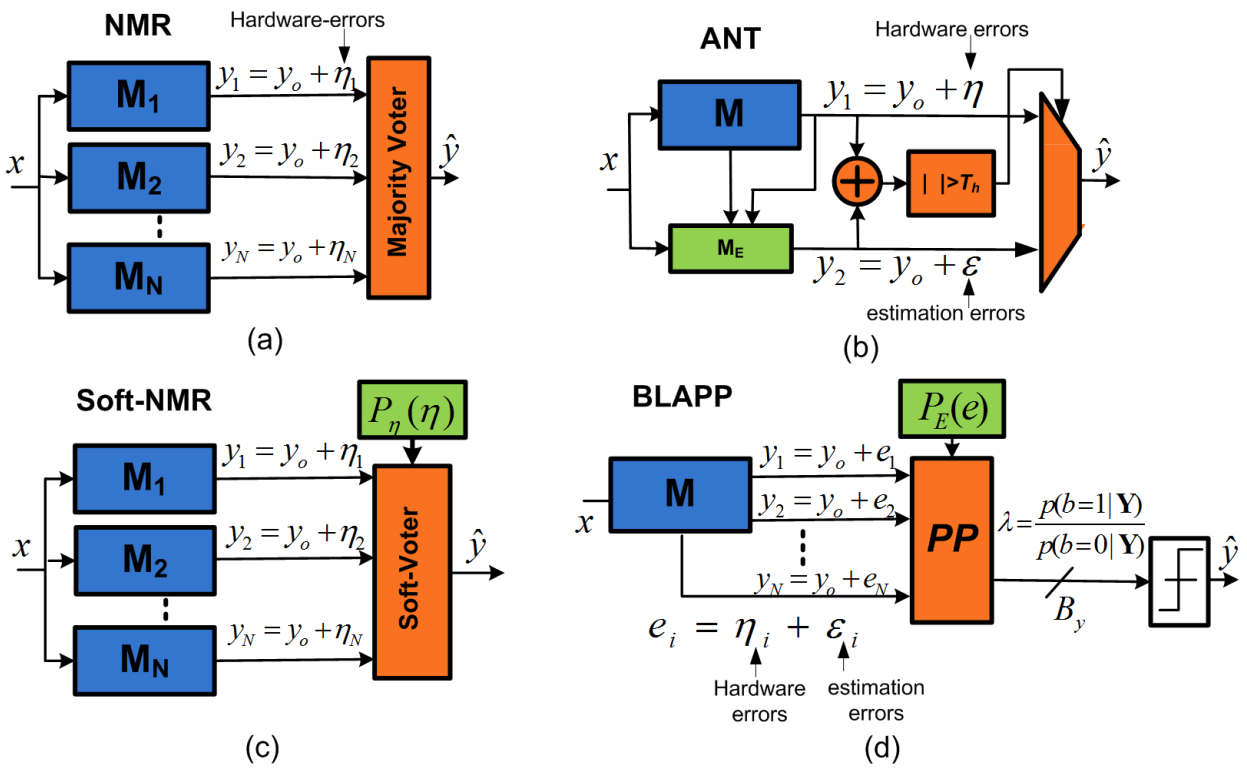


Figure 5.2: Existing architectural level error resiliency techniques: (a) NMR, (b) algorithmic noise-tolerance (ANT), (c) stochastic sensor network-on-chip (SSNOC), and (d) soft NMR.

computational blocks in the form of PMFs has been addressed in [95]. Figure 5.1(c) shows measured error PMF $P_\eta(\eta)$ obtained from a voltage-overscaled IC in a 45-nm CMOS process.

We now describe existing error-resiliency techniques (see Fig. 5.2) using the following definitions:

1. *Observation vector* $\mathbf{Y}$: $\mathbf{Y} = (y_1, y_2, \ldots, y_N)$, where $y_i = y_o + \eta_i + \epsilon_i = y_o + e_i$ with y_o, y_i, η_i, $\epsilon_i \in \mathcal{Y}$.

2. *Decision rule* $\mathcal{R}$: $\hat{y} = f(\mathbf{Y}, P_\eta, P_\epsilon)$, where the corrected/final output $\hat{y} \in \mathcal{Y}$.

For example, NMR (see Fig. 5.2(a)) employs N-way replication, i.e., $e_i = \eta_i$ and $\epsilon_i = 0$, to generate the observation vector $\mathbf{Y}$, and employs majority or other forms of voting strategies as the decision rule $\mathcal{R}$. NMR is effective if the hardware errors η_i's are independent, and is described as follows:

1. $\mathbf{Y}_{NMR} = (y_1, y_2, \ldots, y_N)$, where $y_i = y_o + \eta_i$ with y_o, y_i, $e_i = \eta_i \in \mathcal{Y}$.

2. $\mathcal{R}_{NMR}: \ \hat{y} = maj\left(\mathbf{Y}_{NMR}\right)$, where $maj(.)$ is the majority operator.

Algorithmic-noise tolerance (ANT) [70] employs an estimator block M_E (see Fig. 5.2(b)), which is a low-complexity version of the M-block, to generate an estimate y_2 of the error-free output y_o. The estimator block M_E is designed to be free of hardware errors, i.e., $\eta_2 = 0$ and thus $e_2 = \epsilon$. ANT relies on the difference in the statistics of the hardware error of the M-block η_1 and the estimation error ϵ_2, which can be made small (see $P_\epsilon(\epsilon)$ in Fig. 5.1(b)) to detect and correct for η. Thus, ANT is described as:

1. $\mathbf{Y}_{ANT} = (y_1 = y_o + \eta_1, y_2 = y_o + \epsilon_2)$, where y_1, y_2, η_1, $\epsilon_2 \in \mathcal{Y}$.
2. $\mathcal{R}_{ANT}: \quad \hat{y} = \begin{cases} y_1 & \text{if } |y_1 - y_2| < T_h \\ y_2 & \text{otherwise} \end{cases}$, where T_h is a user defined threshold that maximizes an application-level performance metric.

The stochastic sensor network-on-chip (SSNOC) [74] in Fig. 5.2(c) decomposes the M-block into a set of statistically similar estimator blocks $\mathrm{M}_{E,i}$, and employs robust estimation techniques [75] for error compensation. Thus, SSNOC is described as:

1. $\mathbf{Y}_{SSNOC} = (y_1, y_2, \ldots, y_N)$, where $y_i = y_o + e_i$, $e_i = \eta_i + \epsilon_i$, with y_o, y_i, $e_i \in \mathcal{Y}$.
2. $\mathcal{R}_{SSNOC}: \quad \hat{y} = f_{robust}(\mathbf{Y}_{SSNOC})$, e.g., $f_{robust}(\mathbf{Y}_{SSNOC}) = median(\mathbf{Y}_{SSNOC})$ or $f_{robust}(\mathbf{Y}_{SSNOC}) = huber(\mathbf{Y}_{SSNOC})$.

Soft NMR [78] (see Fig. 5.2(d)) employs the same observation vector as NMR. However, unlike NMR, soft NMR exploits the hardware error PMF $P_\eta(\eta)$, to implement a decision rule $\mathcal{R}$ based on the maximum-likelihood (ML) principle to enhance system robustness. Soft NMR is described as:

1. $\mathbf{Y}_{SNMR} = (y_1, y_2, \ldots, y_N)$, where $y_i = y_o + \eta_i$ with y_o, y_i, $e_i = \eta_i \in \mathcal{Y}$.
2. $\mathcal{R}_{SNMR}: \ \hat{y} = \underset{y_o}{arg\,max}\, P\left(\mathbf{Y}_{SNMR}|y_o\right) = \underset{y_o}{arg\,max}\, P_{\underline{\eta}}(\underline{\eta}|y_o)$, where $\underline{\eta} = (\eta_1, \eta_2, ...\eta_N)$.

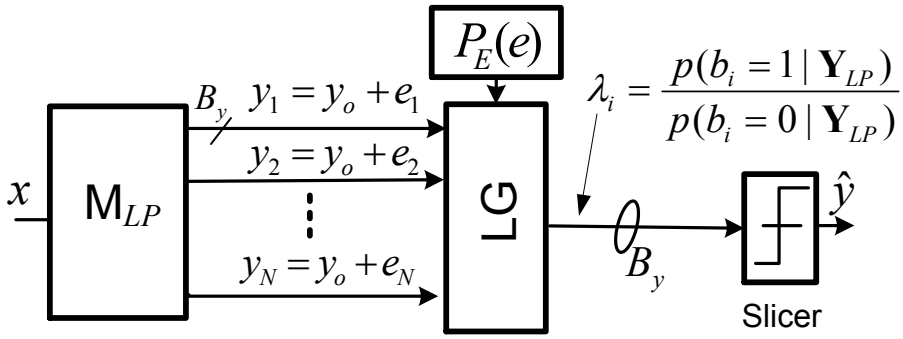


Figure 5.3: The proposed technique: likelihood processing (LP).

Our proposed technique LP (see Fig. 5.3) consists of a computational block M_{LP} generating an N-element observation vector $\mathbf{Y}_{LP}$, a *likelihood generator* (LG), and a *slicer*. Figure 5.4 shows that the block M_{LP} can be designed via one or more of the following techniques: 1) replication, 2) estimation, and 3) exploiting inherent signal correlations in M. In the latter case, intermediate signals from the M are employed to generate $\mathbf{Y}_{LP}$, thereby avoiding any hardware replication (see Fig. 5.4(c)). For example, adjacent pixels in image/video processing applications have correlated values, thereby providing multiple observations with low overhead.

The LG-block in Fig. 5.3 employs the composite error PMF $P_E(e = \eta + \varepsilon)$ to compute the a-posteriori probability (APP) ratio $\lambda_j = P(b_j = 1|\mathbf{Y}_{LP})/P(b_j = 0|\mathbf{Y}_{LP})$ for each output bit b_j $(j = 0, .., B_y - 1)$, of the B_y-bit output y_o. This soft information provides a measure of the confidence/reliability on each output bit. For example, our confidence in bit b_j being a '1' increases with the numerical value of λ_j for $\lambda_j > 1$, and vice versa for bit b_j being a '0'.The slicer in Fig. 5.3 thresholds λ_j to obtain a hard estimate $\hat{b}_j$. In this chapter, we consider only hard decisions for simplicity and ignore the additional improvement available by exploiting soft information further. The LP technique can be described as:

1. $\mathbf{Y}_{LP} = (y_1, y_2, \ldots, y_n)$, where $y_i = y_o + \eta_i + \epsilon_i = y_o + e_i$ with y_o, y_i, $e_i \in \mathcal{Y}$.

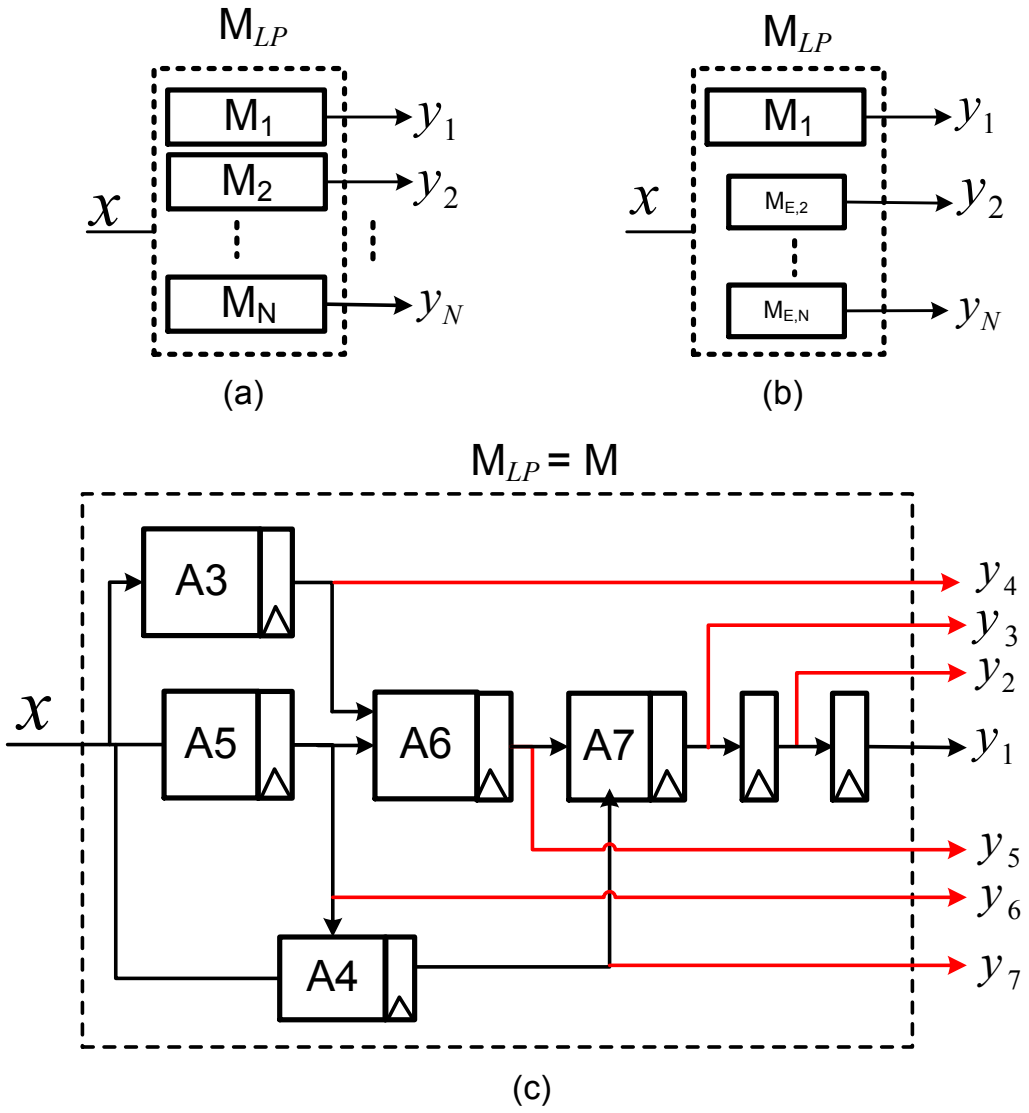


Figure 5.4: Techniques to generate observation vector $\mathbf{Y_{LP}}$: (a) replication, (b) estimation, and (c) spatio-temporal correlation.

2. $\mathcal{R}_{LP}:\ \hat{b_j} = \begin{cases} 1, & \text{if } \lambda_j \geq 1 \\ 0, & \text{otherwise} \end{cases}$, where $\lambda_j = \frac{P(b_j=1|\mathbf{Y}_{LP})}{P(b_j=0|\mathbf{Y}_{LP})}$ for $j = 0, 1, \ldots, B_y - 1$

We employ the notation LPNx-(B_y) to denote the LP technique operating on an observation vector $\mathbf{Y}_{LP}$ of length N and generating a B_y-bit error-compensated output. The character 'x' in LPNx-(B_y) denotes the architectural setup of the M$_{LP}$-block and can take the values 'r', 'e', or 'c' corresponding to the three architectural choices: replication, estimation, or correlation, respectively.

The next section describes the LP framework in detail and shows how to generate the APP ratio λ_j for each output bit b_j.

5.2 The Proposed Technique: Likelihood Processing (LP)

In this section, we present LP in its most general form, illustrate it through an example, and then describe an efficient architecture.

5.2.1 The LP Algorithm

Consider the computational block M in Fig. 5.1(a), whose correct output y_o is represented with B_y bits, $y_o = \{b_j\}_{j=1}^{B_y}$, and manifests an output space $\mathcal{Y} = \{0, 1, \ldots, 2^{B_y} - 1\}$. Furthermore, employing one of the observability enhancing techniques in Fig. 5.4, an observation vector $\mathbf{Y}_{LP} = \{y_i\}_{i=1}^{N}$ is generated, where $y_i = y_o + \eta_i + \varepsilon_i$. The error PMFs $\{P_{E_i}(e_i)\}_{i=1}^{N}$, are assumed to be known as in the case of soft NMR [78].

For each output bit b_j ($j = 1, \ldots, B_y$), we need to compute the APP ratio λ_j defined as follows:

$$\lambda_j = \frac{P(b_j = 1|\mathbf{Y}_{LP})}{P(b_j = 0|\mathbf{Y}_{LP})} = \frac{P(b_j = 1|\mathbf{Y}_{LP})}{1 - P(b_j = 1|\mathbf{Y}_{LP})} \tag{5.2}$$

In practice, computing the log-domain APP ratio Λ_j simplifies the algorithm and implemen-

tation, and is given by:

$$\Lambda_j = \log \lambda_j = \log \frac{P(b_j = 1|\mathbf{Y}_{LP})}{P(b_j = 0|\mathbf{Y}_{LP})} \tag{5.3}$$

Thus, our confidence in b_j being a '1' is very high if $\Lambda_j >> 0$, and vice versa for b_j being a '0'.

Applying the Bayes rule to (5.2), we obtain:

$$P(b_j = k|\mathbf{Y}_{LP}) = \frac{P(\mathbf{Y}_{LP}|b_j = k)P(b_j = k)}{P(\mathbf{Y}_{LP})} \tag{5.4}$$

where $P(\mathbf{Y}_{LP})$ is the probability of observing the vector $\mathbf{Y}_{LP}$ and $k \in \{0, 1\}$. Substituting (5.4) in (5.2), we obtain:

$$\lambda_j = \frac{P(\mathbf{Y}_{LP}|b_j = 1)P(b_j = 1)}{P(\mathbf{Y}_{LP}|b_j = 0)P(b_j = 0)} \triangleq \frac{p_{j,1}}{p_{j,0}} \tag{5.5}$$

where we denote $p_{j,k} = P(b_j = k|\mathbf{Y}_{LP})P(b_j = k)$ for $k = 0$ or 1. For each output bit, b_j, the probabilities $\{p_{j,k}\}_{k=0,1}$ are generated by a bit-level to word-level mapping in the output space $\mathcal{Y}$ as follows:

$$p_{j,k} = \sum_{y_o \in \mathcal{Y}_{j,k}} P\left(\mathbf{Y}_{LP}\,|y_o\right) P\left(y_o\right) \tag{5.6}$$

where $\mathcal{Y}_{j,k} = \{y_o \in \mathcal{Y} \mid b_j = k\}$, i.e., the set of all possible outputs that have the j^{th} bit $b_j = k$, and $P\left(y_o\right)$ is the prior probability, i.e., the distribution of the error-fee output y_o. We assume that the errors e_i are independent in order to reduce the memory complexity inherent in the storage of joint error PMFs. Error independence can be achieved by employing techniques such as data diversity, architecture diversity, or scheduling diversity, which would be described in Chapter 6.

Assuming independent errors, the conditional PMF in (5.6) can be written as:

$$\begin{aligned} P\left(\mathbf{Y}_{LP}\,|y_o\right) &= P\left((y_1, y_2, \ldots, y_N)\,|y_o\right) \\ &= \prod_{i=1}^{N} P\left(y_i\,|y_o\right) = \prod_{i=1}^{N} P\left(e_i = y_i - y_o\right) = \prod_{i=1}^{N} P_{E_i}\left(e_i\right) \end{aligned} \tag{5.7}$$

Therefore, substituting (5.7) into (5.6) results in

$$p_{j,k} = \sum_{y_o \in \mathcal{Y}_{j,k}} \prod_{i=1}^{N} \left[P_{E_i}\left(e_i = y_i - y_o\right)\right] P\left(y_o\right) \tag{5.8}$$

Therefore, substituting (5.8) in (5.5), provides the final expression for λ_j as follows:

$$\lambda_j = \frac{p_{j,1}}{p_{j,0}} = \frac{\displaystyle\sum_{y_o \in \mathcal{Y}_{j,1}} \left[\prod_{i=1}^{N} P_{E_i}\left(e_i = y_i - y_o\right)\right] P\left(y_o\right)}{\displaystyle\sum_{y_o \in \mathcal{Y}_{j,0}} \left[\prod_{i=1}^{N} P_{E_i}\left(e_i = y_i - y_o\right)\right] P\left(y_o\right)} \tag{5.9}$$

Next, we illustrate the LP algorithm with an example.

5.2.2 Motivational Example

Consider the computation kernel M in Fig. 5.5(a) with a B_y = 2-bit output $y = (b_1, b_2)$ corrupted by a 2-bit error $e = (e_1, e_2) = (b_{o,1} \oplus b_1, b_{o,2} \oplus b_2)$ according to the PMF in Fig. 5.5(b), where the *component probability of error* (pre-correction error rate) $p_\eta = p(\hat{y} \neq y_o) = P_E(e \neq (0,0))$. Assume that we rely on a single output observation $y = (b_1, b_2) = (1, 0)$, i.e., $N = 1$, and $\mathbf{Y}_{LP} = (y_1 = (1, 0))$. We refer to this form of likelihood processing as LP1r-(2), i.e., LP with $N = 1$, and $B_y = 2$.

The LG-block computes $p_{1,1}$ in (5.6) by considering all possible outputs $y_o \in \mathcal{Y}_{1,1} = \{y_o =$

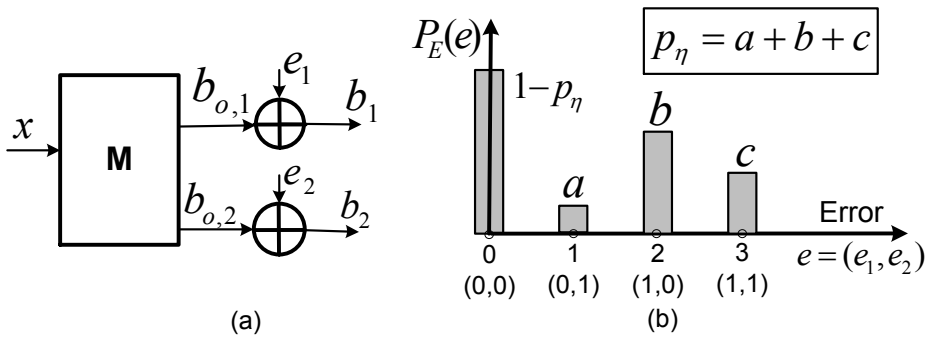


Figure 5.5: An example of LP: (a) 2-bit output erroneous computational block and (b) a 2-bit sample error PMF.

$(1, b_2) \in \mathcal{Y}\}$ as follows:

$$\begin{aligned} p_{1,1} &= \sum_{y_o \in \mathcal{Y}_{1,1}} P\left(\mathbf{Y}_L P \,|y_o\right) P\left(y_o\right) \\ &= P\left(\mathbf{Y}_{LP} \,|y_o = (1,0)\right) P\left(y_o = (1,0)\right) \\ &+ P\left(\mathbf{Y}_{LP} \,|y_o = (1,1)\right) P\left(y_o = (1,1)\right) \end{aligned} \tag{5.10}$$

Assuming all outputs $y_o \in \mathcal{Y}$ are equally likely to occur, we have $P\left(y_o = (1,0)\right) = 0.25$ and $P\left(y_o = (1,1)\right) = 0.25$ in (5.10). Employing the error PMF $P_E(e)$ in Fig. 5.5(b), we write (5.10) as

$$\begin{aligned} p_{1,1} &= P\left(y_1 = (1,0) \,|y_o = (1,0)\right) + P\left(y_1 = (1,0) \,|y_o = (1,1)\right) \\ &= P\left(e = (0,0)\right) + P\left(e = (1,1)\right) \\ &= (1 - p_\eta) + c \end{aligned} \tag{5.11}$$

Assuming that the pre-correction error rate $p_\eta = 0.6$, $a = 0.7p_\eta$, $b = 0.3p_\eta$, and $c = 0$, (5.11) gives $p_{1,1} = 0.4$. Similarly from (5.6), we get $p_{1,0} = a + b = 0.6$. Thus, the APP ratio of first bit $\lambda_1 = 0.67$ and $\Lambda_1 = -0.4$, which is less than 0, and thus the slicer in Fig. 5.3 generates $\hat{b}_1 = 0$. Similarly, one can show that $p_{2,1} = a + c = 0.7p_\eta = 0.42$, $p_{2,0} = (1 - p_\eta) + b = 0.58$, $\lambda_2 = 0.72$ and $\Lambda_2 = -0.33$ hence $\hat{b}_2 = 0$. Thus, the LG-block generates the final sliced error

compensated output $\hat{y} = (0, 0)$ even though the M-block output is $y = (b_1, b_2) = (1, 0)$, i.e., there is a high probability that b_1 is in error, given the knowledge of the error PMF $P_E(e)$.

Next consider LP3r-(2), i.e., we have three identical copies of the M-block in Fig. 5.5(a) generating the observation vector $\mathbf{Y_{LP}} = \{y_1, y_2, y_3\}$ followed by a three-input LG-block and a slicer. Without any loss of generality, we assume that y_1, y_2, and y_3 are corrupted by independent identically distributed (iid) errors e_1, e_2, and e_3, respectively. In other words, the errors e_1, e_2, and e_3 are independent and follow the same error PMF $P_E(e)$ in Fig. 5.5(b), i.e., $P_{E_1}(e_1) = P_{E_2}(e_2) = P_{E_3}(e_3) = P_E(e)$. If the observation vector $\mathbf{Y}_{LP} = (y_1 = (1, 0), y_2 = (1, 0), y_3 = (0, 1))$, then TMR selects $\hat{y} = (1, 0)$ via a majority vote. On the other hand, a smart voter with the knowledge of error statistics would realize that the correct output $y_o \neq (1, 0)$ since $e_3 = y_3 - y_o = (1, 1)$ but $P_{E_3}(e_3 = (1, 1)) = c = 0$ (see Fig. 5.5(b)). For example, employing LP3r-(2), the computation of $P(\mathbf{Y}_{LP}|y_o = (1, 1))$ in (5.6) is given by:

$$\begin{aligned} P\left(\mathbf{Y}_{LP}|y_o = (1,1)\right) &= P\left(e_1 = (0,1), e_2 = (0,1), e_3 = (1,0)\right) \\ &= P\left(e_1 = (0,1)\right)P(e_2 = (0,1))P(e_3 = (1,0)) \\ &= (ap_\eta)^2(bp_\eta) = ba^2p_\eta^3 \end{aligned}$$

Similarly, $P\left(\mathbf{Y}_{LP}|y_o = (1, 0)\right) = ap_\eta(1 - p_\eta)^2$. Assuming $p_\eta = 0.5$, $a = 0.7p_\eta$, $b = 0.3p_\eta$, $c = 0$, and equal priors, (5.10) results in $P(b_1 = 1|\mathbf{Y}_{LP}) = 0$, i.e., $\lambda_1 = 0$ indicating our confidence in the decision $\hat{b}_1 = 0$ is very high. Similarly, one can show for the second bit b_2, $p_{2,1} = (cp_\eta)^2(1 - p_\eta) = 0.07$, $p_{2,0} = (bp_\eta)^2(cp_\eta) = 0.019$, $\lambda_2 = 3.4$, and $\Lambda_2 = 1.22$ and hence $\hat{b}_2 = 1$, resulting in the corrected output as $\hat{y} = (0, 1)$. We also observe that the log-domain APP ratio moved farther away from 0 with $N = 3$ than that with $N = 1$. This indicates that multiple observations increase our confidence in the decision $\hat{y} = (0, 1)$. The effectiveness of LP over conventional design can be calculated for this example by injecting errors at the output of 2-bit computational kernel(s) according to the PMF in Fig. 5.5(b) at various

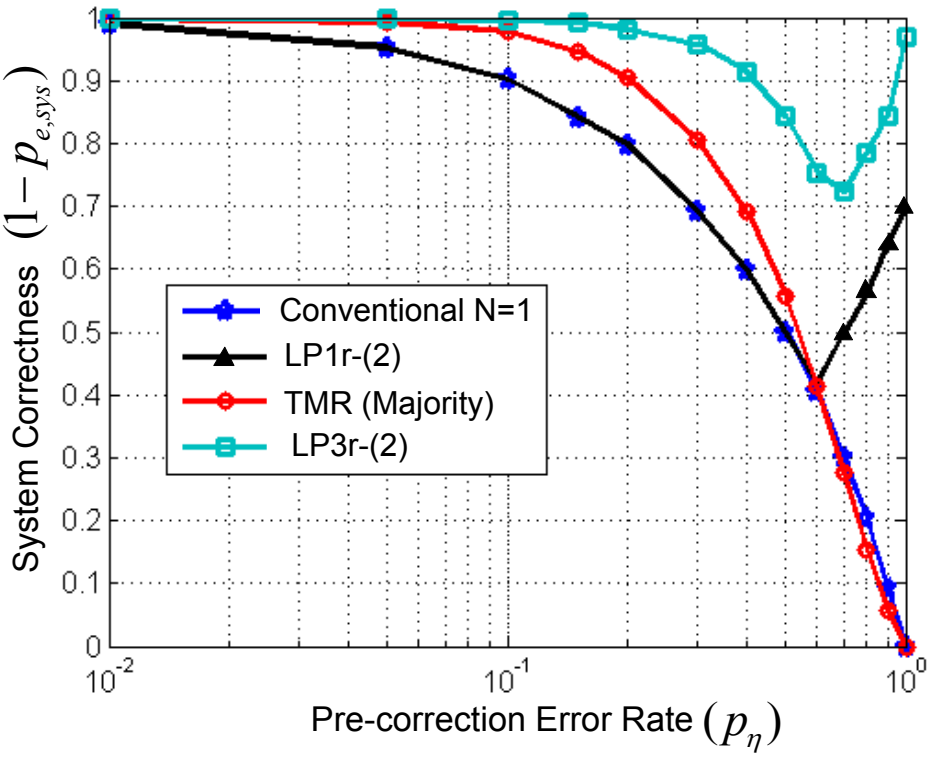


Figure 5.6: System correctness of a 2-bit output system at different p_η.

pre-correction error rates p_η's. LPNr-(2) employs $P_E(e)$ to generate probabilities for b_1 and b_2 from $\mathbf{Y_{LP}}$ followed by a slicer to produce a hard estimate $\hat{y}$, while a conventional design directly uses $\mathbf{Y_{LP}}$ followed by a majority vote. The *system correctness metric*, defined as $P(\hat{y} = y_o) = 1 - P(\hat{y} \neq y_o) = 1 - p_{e,sys}$, is employed to compare LP to conventional design.

From Fig. 5.6, we observe that LP3r-(2) outperforms TMR for all values of p_η. Second, system correctness for both LP1r-(2) and LP3r-(2) increases with p_η for $p_\eta \geq 0.6$ and $p_\eta \geq 0.7$, respectively. This unusual outcome occurs because LP understands that the observations in $\mathbf{Y}_{LP}$ are unreliable for high values of p_η, and thus tends to choose outputs from $\mathcal{Y}$ that do not belong to $\mathbf{Y}_{LP}$. Note that when $p_\eta \geq 0.6$, system correctness for TMR falls below even LP1r-(2) and the conventional $N = 1$ system, because the probability of two or more identical errors becomes larger, and hence the majority voter selects the wrong values more

often. On the other hand, LP exploits the knowledge of the error distribution in Fig. 5.5(b), i.e., different error magnitudes have different error probabilities, to correct for errors.

5.2.3 The N-Input Likelihood Generator (LG-Block) Architecture

We use the log domain processing of probabilities to simplify the implementation of the LG-block in Fig. 5.3. Taking the logarithm (base 2) of (5.8), we obtain:

$$\log p_{j,k} = \log \sum_{y_o \in \mathcal{Y}_{j,k}} 2^{\left[\sum_{i=1}^{N} \log P_{E_i}(e_i = y_i - y_o)\right] + \log P(y_o)} \tag{5.12}$$

We use the *log-max* approximation [96] to simplify (5.12). The *log-max* approximation is given by:

$$\log\left(2^x + 2^y\right) = \max(x, y) + C(|x - y|) \approx \max(x, y) \tag{5.13}$$

where $C(.)$ is a correction term. Thus, ignoring the correction term in (5.13), (5.12) can be written as:

$$\begin{aligned} \log p_{j,k} &\approx \max_{y_o \in \mathcal{Y}_{j,k}} \left[\sum_{i=1}^{N} \log P_{E_i}\left(e_i = y_i - y_o\right) + \log P\left(y_o\right)\right] \\ &= \max_{y_o \in \mathcal{Y}_{j,k}} \left[\Gamma(y_o) + \log P\left(y_o\right)\right] \end{aligned} \tag{5.14}$$

where $\Gamma(y_o)$ is referred to as the *word-metric* and is defined as:

$$\Gamma(y_o) = \sum_{i=1}^{N} \log P_{E_i}\left(e_i = y_i - y_o\right) \tag{5.15}$$

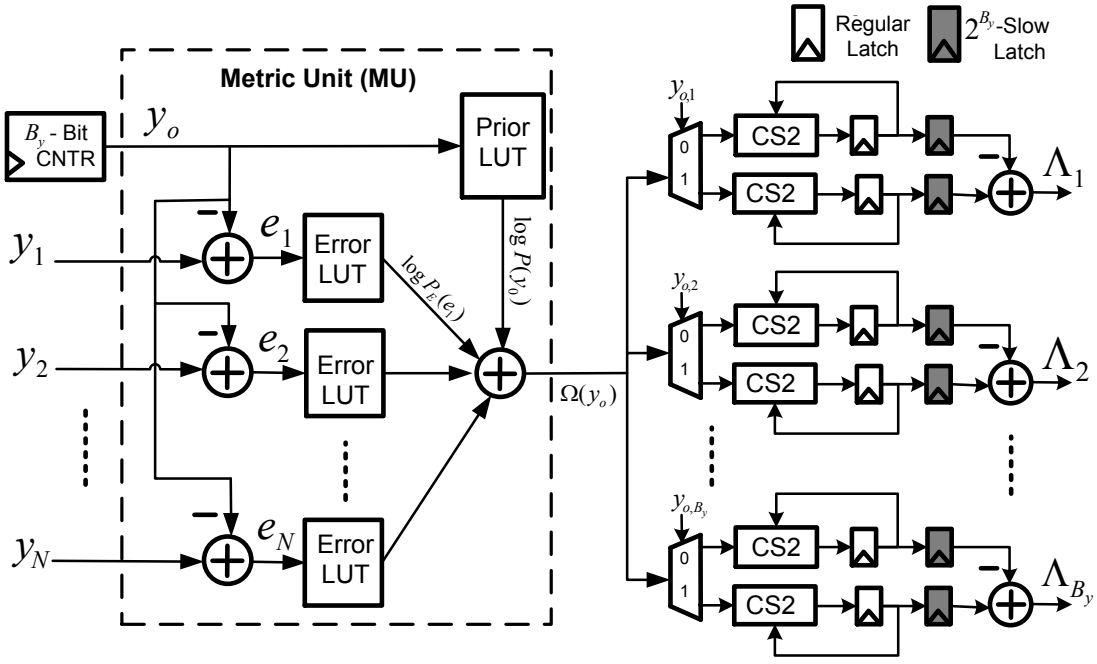


Figure 5.7: An LG-processor architecture for LPN-(B_y) (MU: metric unit and CS2: 2-operand compare-select unit).

From (5.5) and (5.14), the log-APP ratio Λ_j for bit b_j is computed as follows:

$$\begin{aligned}\Lambda_j &= \log p_{j,1} - \log p_{j,0} \\ &\approx \max_{y_o \in \mathcal{Y}_{j,1}} [\Gamma(y_o) + \log P(y_o)] - \max_{y_o \in \mathcal{Y}_{j,0}} [\Gamma(y_o) + \log P(y_o)] \\ &= \max_{y_o \in \mathcal{Y}_{j,1}} \Omega(y_o) - \max_{y_o \in \mathcal{Y}_{j,0}} \Omega(y_o) \end{aligned} \tag{5.16}$$

where $\Omega(y_o) = \Gamma(y_o) + \log P(y_o)$, and $\Gamma(y_o)$ is given by (5.15).

The LG architecture implementing (5.16) is shown in Fig. 5.7. The look-up tables (LUTs) in Fig. 5.7 store the output prior ($\log P(y_o)$), and error PMFs ($\log P_E(e)$). The LG-block generates Λ_j after 2^{B_y} clock-cycles. In each clock cycle, a specific $y_o \in \mathcal{Y}$ is fed into the metric unit (MU). The MU compares y_o to each of the N observations y_i and generates $\Omega(y_o)$. For each b_j, there are two recursive compare-select (CS) units to keep track of the

Table 5.1: Complexity of an L-parallel LG-processor for LPNx-(B_y).

Parallelization Factor	Latency	Storage	Computational Complexity	Activation Factor
$L \leq 2^{B_y}$	$\frac{2^{B_y}}{L}$	$2\left(2^{B_y} \times B_p\right)$ bits	$2L \times N + L + B_y$ Add, and B_y $(log_2 L + 2)$ CS2	α_{LP} $= 1 - \prod_{i=1}^{i=N} (1 - p_{\eta,i})$

maximum values of $\Gamma(y_o) + \log P(y_o)$ computed over $\mathcal{Y}_{j,1}$ and $\mathcal{Y}_{j,0}$, respectively, according to (5.16).

Complexity and Power Overhead

The complexity of the LG-block depends only on the output precision (B_y) and the number of observations N, and is independent of the complexity of the main M$_{LP}$-block complexity. Thus, as the complexity of M$_{LP}$ increases, LPNx-(B_y) overhead constitutes a smaller portion of the total system complexity, resulting in higher energy and robustness benefits. The **LG** architecture in Fig. 5.7 needs 2^{B_y} clock-cycles to compute Λ_j. Parallelization by a factor of $L \leq 2^{B_y}$ reduces the number of clock-cycles to $2^{B_y}/L$ but increases hardware complexity. Complexity estimate of an L-parallel LG-block operating on B_y bits is shown in Table 5.1. The storage of prior and error PMFs (P_E and $P(y_o)$) requires storing $2(2^{B_y} \times B_p)$ bits where it is assumed P_E and $P(y_o)$ are quantized to B_p bits.

5.2.4 Low-Complexity LP Architectures

The complexity and power overhead of LP can be reduced significantly by *probabilistic activation* and *bit-subgrouping* as explained next.

Probabilistic Activation

The LG-block in LP can be activated only when there is a large difference between the observations y_i, which indicates the presence of a large error (see the activation block in Fig. 5.8). Assuming hardware errors are large and independent across the observations y_i,

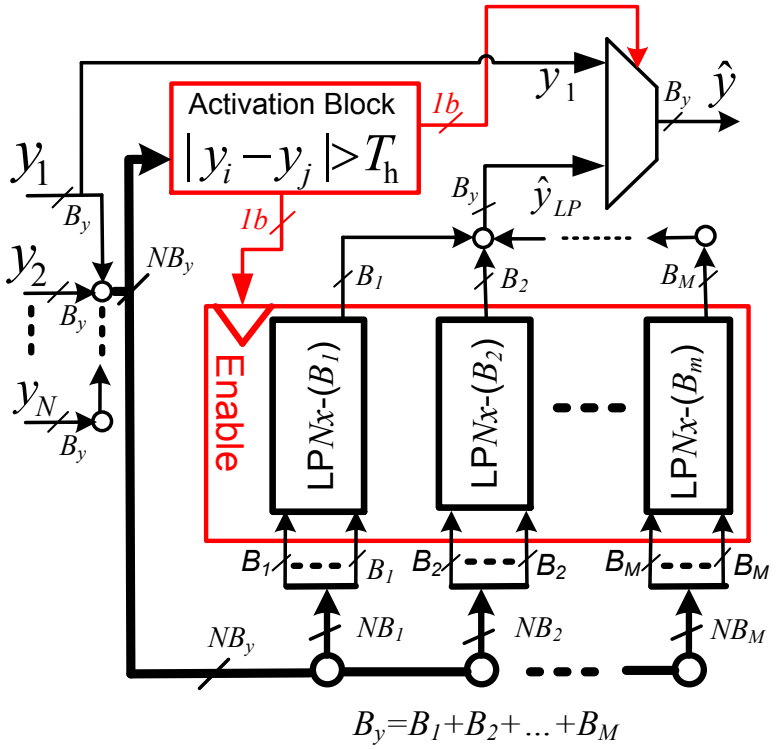


Figure 5.8: A bit-subgrouped LG-processor architecture for LPNx-(B_y) (LPNx-($B_1, B_2, ..., B_m$)) with probabilistic activation module.

then the LG-block activation factor, α_{LP}, is given by:

$$\begin{aligned} \alpha_{LP} &= 1 - P\left(|y_i - y_j| \leq T_h \;\; for\; all (i,j)\; such\; that\; i,j \in \{1,2,...,N\}\; and\; i \neq j\right) \\ &\approx 1 - \prod_{i=1}^{i=N} (1 - p_{\eta,i}) \end{aligned} \tag{5.17}$$

where $p_{\eta,i}$ is the probability of hardware error in y_i.

Bit-Subgrouping

Since the output search space $\mathcal{Y}$ is exponential in the number of output bits B_y, the complexity and power of the LG-block can be reduced via *bit-subgrouping*, as shown in Fig. 5.8. In bit-subgrouping, the B_y-bit output is divided into m subgroups with precisions B_1, B_2,...,B_m,

respectively, such that $B_y = B_1 + B_2 + ... + B_m$. Then, LP is applied independently on each subgroup of B_i bits. With bit-subgrouping, LPNx-(B_y) is denoted as LPN-($B_1, B_2, \ldots, B_m$). Bit-subgrouping significantly reduces the output search space, and thus the LP overhead. For example, if B_y-bit output is uniformly divided into m subgroups each with $B_i = B_y/m$ bits, m LPNx-(B_y/m)'s are needed instead of a single LPNx-(B_y), thereby reducing the storage and computational complexity space of the LG-block from 2^{B_y} to $m2^{B_y/m}$. However, as m increases, the system-level correctness p_{sys} or the robustness of LP will be reduced, since bit-subgrouping ignores error correlations between adjacent subgroups of bits.

5.3 Simulation Results

We demonstrate benefits of LP in terms of robustness and energy efficiency in the design of a two-dimensional inverse discrete-cosine transform (2D-IDCT) image codec subject to PVT errors. The DCT-IDCT transform (see Fig. 5.9(a)) is applied on 256×256 8-bit pixel images, stored initially in memory (Mem), in blocks of 8×8 pixels using Chen's algorithm [97]. Each 2D transform employs two 1D transforms: the first is applied column-wise on the input block, and the second is applied row-wise on the output of the first. Transposition memory (TM) is used to swap the data between rows and columns. The quantizer (Q) and inverse quantizer (Q^{-1}) employ the JPEG quantization table for compression. Only the receiver computational kernels (Q^{-1} and IDCT blocks) are subject to hardware errors. The error-free DCT-IDCT codec achieves a peak signal-to-noise ratio (PSNR) of 33 dB, where the PSNR is defined as

$$PSNR = 10log_{10}(\frac{(255)^2}{E[(y_o - \hat{y})^2]}) \tag{5.18}$$

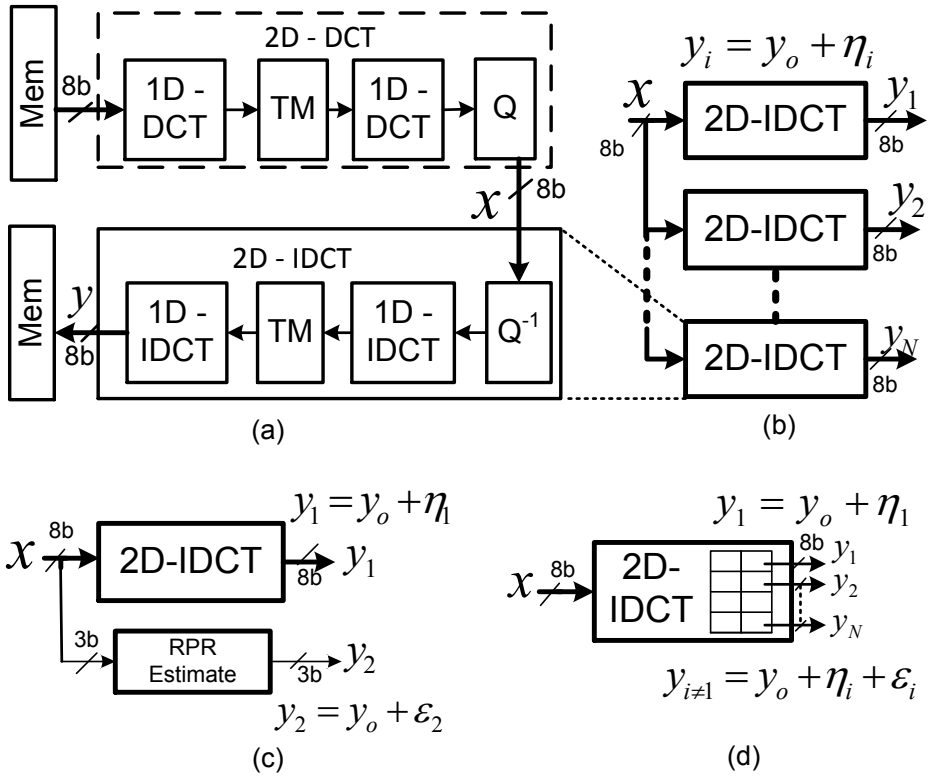


Figure 5.9: An 8-bit output 2D-DCT/IDCT codec: (a) single codec, (b) replication set-up, (c) estimation set-up, and (d) spatial correlation set-up.

5.3.1 System and Architecture Set-up

We employ three different setups to generate multiple observations of the 8-bit output pixel in order to detect and correct hardware errors in the 2D-IDCT block:

1. Replication (see Fig. 5.9(b)): 2D-IDCT kernel is replicated to provide exact estimates of y_o corrupted by hardware errors η_i only. Such a setup is typical of robust general computing systems. Data, architecture, and scheduling diversity can be employed to ensure error independence across redundant kernels, as will be discussed in Chapter 6. Such a setup can be employed in ULP sensing platforms for critical applications where complexity can be traded for increased robustness.

2. Estimation (see Fig. 5.9(c)): A reduced-precision redundancy (RPR) estimator of the

2D-IDCT is employed in parallel with the main 2D-IDCT. While the main block is designed to operate on 8-b pixel inputs, the RPR estimator is designed to operate on the three most-significant bits (MSB) of the pixel value, and thus it is of lower complexity allowing, it to be designed hardware error-free at low overhead. The estimator output y_2 is corrupted by estimation error ϵ_2 only.

3. Spatial correlation (see Fig. 5.9(d)): Application-level data correlations are exploited to generate multiple observations at very low overhead, thereby avoiding approximate or full replication overhead. In the IDCT output, pixels in adjacent rows have similar values and have independent errors since the 1D-IDCT is applied row-wise on the image in the final step of the 2D-IDCT. Thus, pixels in adjacent rows are employed to generate multiple observations y_i. For example, a 4-element observation vector $\mathbf{Y_{LP}}$ for the pixel at row and column coordinates $y_1 = (r_j, c_j)$ is generated by choosing pixels $y_1 = (r_j, c_j)$, $y_2 = (r_j - 1, c_j)$, $y_3 = (r_j - 2, c_j)$, and $y_4 = (r_j + 1, c_j)$. Thus, pixels in the observation vector other than y_1 are corrupted by both estimation and replication errors, while y_1 is corrupted only by estimation error.

Error correction mechanisms, such as NMR (Fig. 5.2(a)), ANT (Fig 5.2(b)), and soft NMR (Fig. 5.2(d)), and the proposed technique LP (Fig. 5.3) are employed to process the observation vector and correct for errors in each setup. All error detection and correction blocks are operated at a critical supply voltage of ($V_{dd-crit} = 0.7\,\mathrm{V}$), to ensure correct operation while consuming minimum power. LP employs a 2^{B_y}-parallel version of the **LG**-processor in Fig. 5.7 so that it requires one clock-cycle to generate the output APP ratios. Different bit-subgroupings of the 8-bit output (see Fig. 5.8) are applied to study the trade-off between LP complexity overhead and performance. In addition to that, a probabilistic activation module for LG-block (see Fig. 5.8) is employed to decrease LP power overhead. The error and prior PMFs are quantized to 8-bits before being stored by LG-block. The complexity of different blocks in error-compensated 2D-IDCTs is shown in Table 5.2.

Table 5.2: Gate complexity (normalized to NAND2) of building blocks in error-compensated 2D-IDCT architectures.

8-bit 2D-IDCT Module	3-bit RPR Estimator	TMR 2D-IDCT Module	N=3 Majority Voter
64.2 k	20.4 k	192.5 k	0.13 k
ANT Compare-Select Module	LG-processor for LP3x-(8)	LG-processor for LP3x-(5,3)	LG-processor for LP3x-(1,1,...1)
0.22 k	50.8 k	14.6 k	0.6 k

5.3.2 Error Characterization and Simulation Procedure

Likelihood processing requires statistical characterization of output errors of the computation engine (DCT-IDCT codec). Accurate modeling of errors increases resiliency at the expense of increased storage requirement and search space of LP. Generally, hardware errors are a function of the PVT settings, the input space, and the architecture. In Chapter 6, we will show that timing error statistics are relatively independent of the input statistics and are a strong function of the architecture. Hence, a training input data I_T can be employed to statistically characterize the error statistics of the M-block. This captures the dependence of hardware errors on the architecture, indirectly considers the dependence on the input statistics, and provides the LG-block with a good estimate of the actual output errors. This training phase can be performed either off-chip or on-chip.

We evaluate the robustness and energy efficiency of the proposed soft-output technique under timing violations caused by PVT variations. As mentioned earlier, VOS is used to emulate these timing violations using a 45-nm TI CMOS process. Keeping the frequency fixed, the supply voltage V_{dd} is reduced beyond a critical design voltage $V_{dd-crit}$ so that intermittent timing errors appear. The simulation methodology involves two steps:

1. **Training phase**: An error PMF $P_E(e)$ is obtained via an RTL simulation of the M-block employing a *training* input data-set I_T as follows:

    - Step 1: Circuit simulations are employed to characterize the worst-case delay vs. V_{dd} relationship of the gate library.

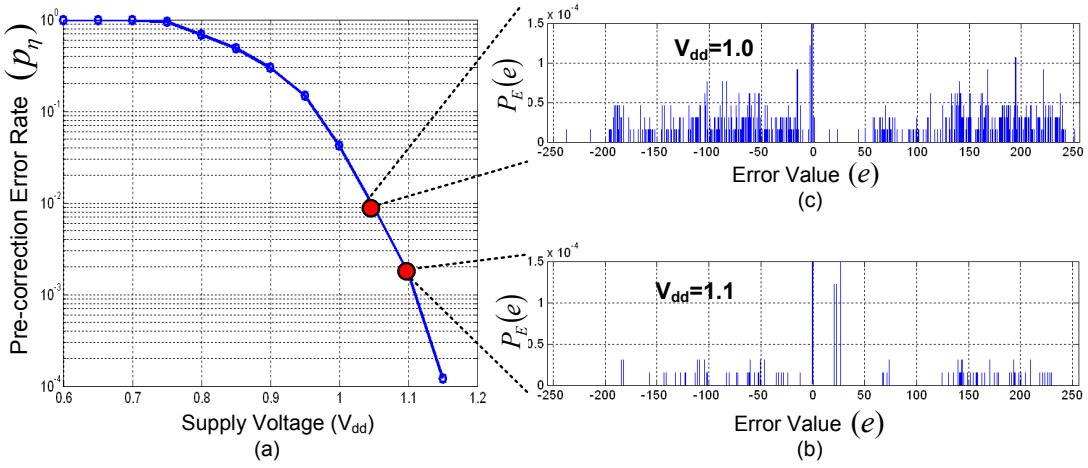


Figure 5.10: VOS errors in 2D-IDCT: (a) pre-correction error rate (component probability of error) p_η, and output error PMFs ($P_E(e)$) at (b) $V_{dd} = 1.1\,\text{V}$ and (c) $V_{dd} = 1\,\text{V}$.

- Step 2: At each $V_{dd} < V_{dd-crit}$, a gate-level netlist of the M-block is simulated using gate delays obtained from Step 1 and employing the data-set I_T while the frequency of operation is fixed to meet $V_{dd-crit}$ timing constraints only. This step generates the erroneous output y in (5.1).
- Error PMFs $P_E(e)$s are obtained at each V_{dd} by comparing y_o and y as shown in (5.1).

2. **Operational phase**: The M-block operates under VOS on an actual data-set I_A, which is different from I_T, and exhibits I_A-dependent errors. LP employs the I_T-dependent error PMF $P_E(e)$ obtained during the training phase for error compensation.

Figure 5.10(a) shows the pre-correction error rate p_η at the output of 2D-IDCT as V_{dd} is reduced from 1.2 V to 0.6 V. Each point on the curve is characterized by an error PMF $P_E(e)$. For example, error PMFs for the 8-bit output 2D-IDCT at 1.1 V and 1.0 V are shown in Figs. 5.10 (b) and (c), respectively. As voltage is reduced, more spread in error values is observed because more circuit paths begin to fail to meet the timing constraints.

5.3.3 System Performance and Robustness

The PSNR for DCT-IDCT codec output under replication (Fig. 5.9(b)) is shown in Fig. 5.11 for different p_η (probability of output error for a single 2D-IDCT) corresponding to different V_{dd}. Figure 5.10(a) is employed to relate V_{dd} to p_η. Figure 5.11(a) shows that LP3r-(8) can tolerate 70×, 5× and 3× higher pre-correction error rate p_η compared to the conventional (uncompensated) single 2D-IDCT, TMR, and soft TMR, respectively, in order to achieve a PSNR of 30 dB. Interestingly, as seen in Fig. 5.11(a), LP2r-(8), i.e., LP with dual-MR (DMR), behaves close to or even better than TMR when $p_\eta \geq 0.05$, i.e., LP with a replication factor of two outperforms TMR. This is unlike conventional DMR, which can only detect errors but not correct them. The effect of bit-subgrouping on LP3r performance is studied in Fig. 5.11(b). Bit-subgrouping of LP3r-(8) to LP3r-(5,3), where one sub-LP operates on the 5-MSBs and the second on the rest of the 3-LSBs, minimally affects the robustness of LP3r-(8). As bit-subgrouping increases in order to decrease LP overhead, the robustness benefits of LP over TMR reduce as expected. This is because more error correlations across adjacent bits are being ignored. However, even with 1-bit sub-grouping (LP3r-(1,1,...,1)) LP still outperforms TMR.

The PSNR of the DCT-IDCT codec under estimation setup (Fig. 5.9(c)) is shown in Fig. 5.12(a) for different pre-correction error rates p_η's. The 3-bit RPR estimator is not subject to VOS. The estimator alone achieves a PSNR of 22.2 dB. LP2e-(8) processes the two outputs y_1 (main block) and y_2 (estimator) similarly to ANT (see Fig. 5.9(c)), and achieves robustness enhancement of 100× and 5× compared to the uncompensated single 2D-IDCT, and ANT, respectively, at a PSNR of 30 dB. LP2r-(8) robustness benefits reduce with bit-subgrouping (LP2e-(5,3)) and LP becomes less efficient than ANT.

The PSNR of the DCT-IDCT codec under the spatial-correlation setup (Fig. 5.9(d)), where adjacent row pixels are used as estimators, is shown in Fig. 5.12(b). Only LP with (5,3) bit-subgrouping is shown. Similarly to the case of replication, simulations confirm that

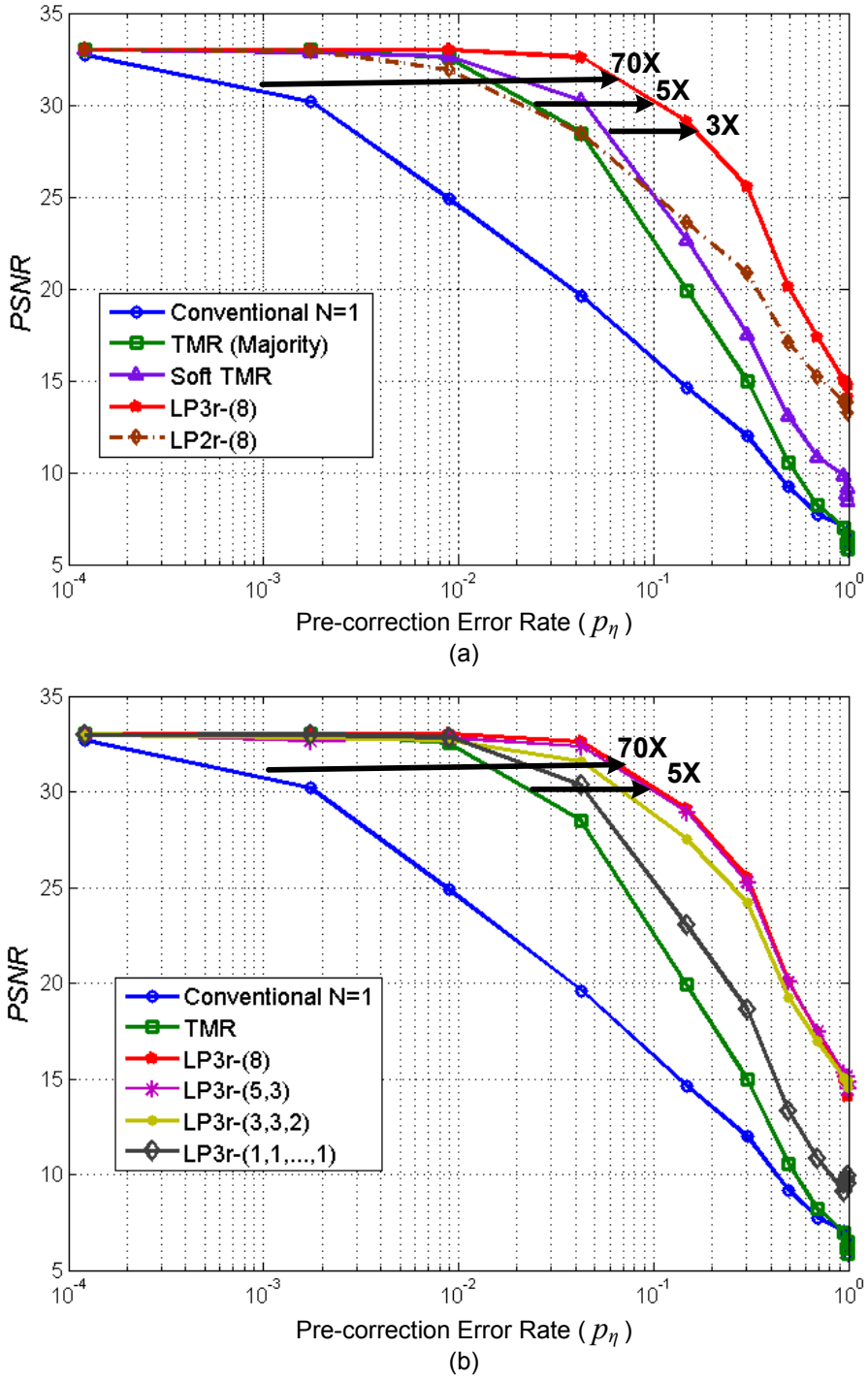


Figure 5.11: System robustness of 2D DCT-IDCT codec under replication: (a) comparing LPNr-(8) to other error-resilient techniques without bit-subgrouping and (b) LPNr-(8) performance with bit-subgrouping.

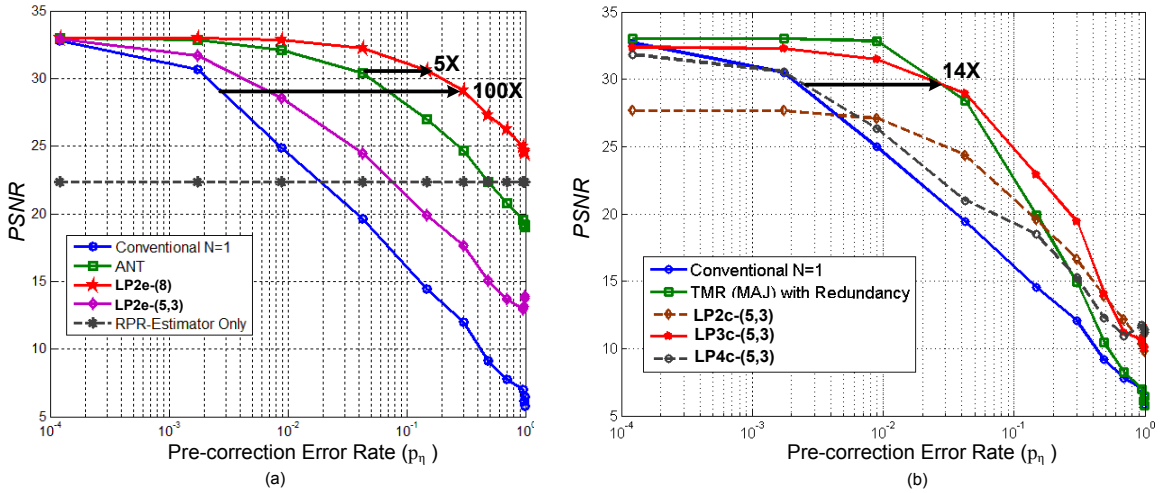


Figure 5.12: System robustness of 2D DCT-IDCT codec using (a) estimation and (b) spatial correlation.

bit-subgrouping under correlation setup from (8) to (5,3) shows negligible loss in performance. In Fig. 5.12(b), LP3c-(5,3) uses pixels in adjacent two rows as estimators for the current pixel and achieves 14× increase in robustness at a PSNR of 30 dB compared to the conventional system. This level of robustness is similar to that achieved by TMR. However, the latter has two additional M-blocks (IDCTs) compared to LP3c-(5,3). Note that, majority voting does not apply for the spatial correlation setup, since the multiple observations are corrupted by estimation errors when they are hardware error-free. Using two (current and previous-row) instead of three adjacent pixels, LP2c-(5,3) shows worse performance than LP3c-(5,3), since it has a smaller number of estimators. In fact, it behaves worse than conventional design when $p_\eta < 0.004$ because for $p_\eta < 0.004$, estimation errors dominate hardware errors and LP2c-(5,3) is not able to determine which of the two outputs is correct. LP4c-(5,3) performance also degrades compared to LP3c-(5,3) because LP4c-(5,3) employs pixels that are spatially farther apart than LP3c-(5,3), leading to higher estimation error. Thus, the performance of LPNc-(5,3) depends upon the relative contribution of estimation and hardware errors to the PSNR.

The perceptual quality of a sample image under different error-compensation techniques

Figure 5.13: Sample codec output images: (a) original image, (b) error-free IDCT ($p_\eta = 0, PSNR = 33\,\text{dB}$), (c) erroneous single IDCT ($p_\eta = 0.13, PSNR = 14\,\text{dB}$), (d) majority-vote TMR ($p_\eta = 0.13, PSNR = 19\,\text{dB}$), (e) LP3c-(5,3) ($p_\eta = 0.14, PSNR = 24\,\text{dB}$), (f) ANT ($p_\eta = 0.13, PSNR = 26\,\text{dB}$), (g) LP3r-(5,3) ($p_\eta = 0.13, PSNR = 29\,\text{dB}$), (h) LP2e-(8) ($p_\eta = 0.13, PSNR = 31\,\text{dB}$).

is shown in Fig. 5.13 where the underlying hardware has the same pre-correction error rate $p_\eta = 0.13$. TMR achieves only 5 dB improvement in PSNR over the conventional system, thereby failing to improve the image quality significantly. LP3r-(5,3) achieves much better image quality with a PSNR of 28 dB. This result corresponds to a 14 dB improvement in PSNR over the conventional system. Using the RPR-estimator setup, LP3e-(8) achieves the best image quality (PSNR of 31 dB) with just a very few noticeable errors in the image. Avoiding any form of redundancy and using only signal correlations, LP3c-(5,3) achieves relatively much better quality than TMR with 9 dB instead of 5 dB improvement in PSNR over the conventional system.

Therefore, we see that LP provides a tremendous increase in robustness to circuit errors at the same level of application metric (output quality) compared to conventional systems.

This translates to significant improvement in the application metric at the same degree of unreliability in the circuit fabric. Under exact replication and estimation (approximate replication), LP outperforms existing error-resiliency techniques such as NMR and ANT in terms of robustness and application metric. It can also provide robustness and quality levels similar to those of NMR and ANT while avoiding any form of exact or approximate replication.

5.3.4 Power Savings

HSPICE is used to estimate the power consumption of the gate library at different V_{dd}'s in the 45-nm TI CMOS process. The total power for the computational kernel (2D-IDCT) and error-compensation blocks (majority voter, soft voter, RPR estimator, and G-processors) is obtained by summing up the individual power of constituent gates under various architectural setups (replication, estimation, and correlation). Since the LG-processor is activated only when a large difference is observed among the observations (see Fig. 5.7(b)), the LG-processor power overhead is scaled by its probabilistic activation factor. Figure 5.14 shows the power consumption at each PSNR in Fig. 5.11(a) and Figs. 5.12 (a) and (b) for the three different setups. The conventional ($N = 1$) architecture is error-free only at $PSNR = 33\,\text{dB}$.

In the case of replication (see Fig. 5.14(a)), LP3r-(5,3) achieves 15% power savings compared to TMR for a wide range of PSNR. These power savings are achieved with additional robustness to hardware error, i.e., higher component probability of error (pre-correction error rate), at the same PSNR as illustrated in Fig. 5.11(a). We can trade this additional robustness for further power savings. For example, at a PSNR of 28 dB, LP2r-(8) tolerates the same level of component probability of error as TMR in Fig. 5.11(a) but achieves a power savings of 35% (see Fig. 5.14(a)). Bit-subgrouping of LP from (8) to (5,3) increases the power savings at a given PSNR due to the reduced complexity overhead. However, LP3r-(8)

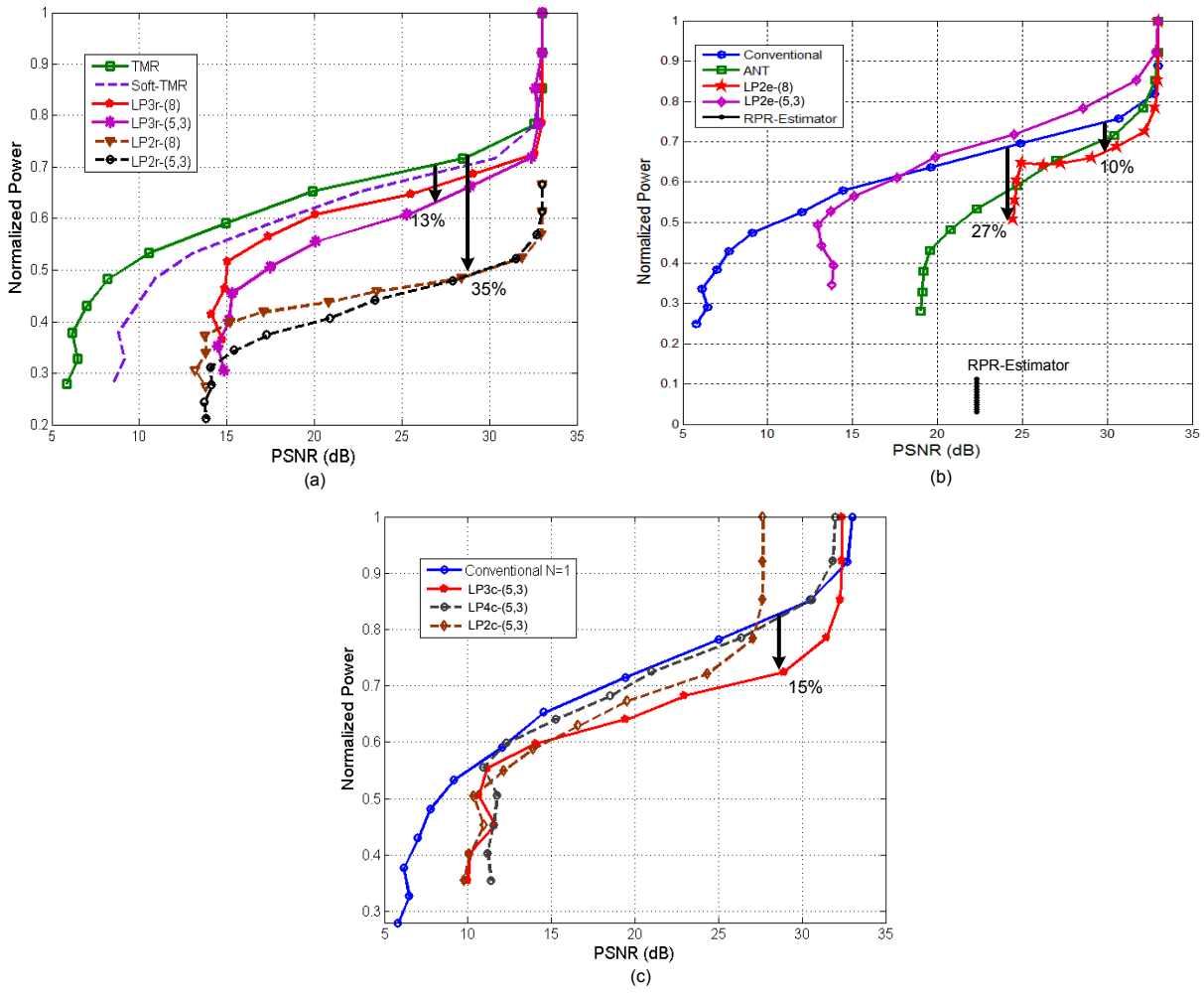


Figure 5.14: LP power savings in a 45-nm TI CMOS process: (a) replication, (b) estimation, and (c) spatial-correlation setups.

achieves slightly better robustness at the same PSNR compared to LP3r-(5,3) as illustrated in Fig. 5.11(b).

In the case of estimation setup (see Fig. 5.14(b)), LP2e-(8) power savings compared to conventional design ranges between 10% and 27% for different PSNR and are slightly better than the ANT-based design. This is in addition to the 100× improvement in robustness over the conventional design. Note that bit-subgrouping in the estimation setup results

in additional power overhead at the same PSNR, since its robustness loss in Fig. 5.12(a) compared to LP2e-(8) is significant compared to the logic complexity savings it provides.

In the spatial-correlation setup (see Fig. 5.14(c)), LP3c-(5,3) achieves 15% power savings compared to conventional designs. If we want to trade increased robustness provided by LP for additional power savings, TMR will employ two more IDCT modules to achieve similar robustness to LP3c-(5,3) at 28 dB of PSNR (see Fig. 5.12(b)). In this case, the power savings of LP3c-(5,3) will be 71% compared to TMR, since LP3c-(5,3) uses two fewer IDCT modules.

5.4 Summary

We presented a stochastic computing technique referred to as *likelihood processing* to design robust systems by exploiting error statistics at bit-level. Techniques from detection and estimation theory, in particular the maximum a-posteriori (MAP) rule, are employed to generate reliability information or confidence-level on each output bit enabling the correction of errors in an optimal probabilistic sense. Simulations in DCT image codec show that LP improves on existing reliable system design techniques such as TMR, as well as on stochastic computing techniques such as ANT and soft NMR. Energy savings up to 71% and robustness benefits up to 100× are illustrated.

CHAPTER 6

CHARACTERIZATION AND ENGINEERING OF TIMING ERROR STATISTICS FOR STOCHASTIC COMPUTING PLATFORMS

Chapter 5 demonstrated the benefits of employing error statistics for robust and energy-efficient DSP kernels in emerging ULP applications. It is clear that the availability of statistical error models, and developing an understanding of the factors that impact error statistics are essential in the design of stochastic computing systems. Furthermore, the availability of error statistics enables robustness analysis of existing techniques, as done in [77] and [98] for NMR, ANT, and soft NMR. However, error modeling and abstraction is a hard problem because errors, in particular timing errors, are a function of a number of parameters such as the input statistics, path delay distribution of the architecture, PVT corner, and other physical parameters.

This chapter makes a case for developing statistical timing error models of DSP kernels implemented in nanoscale circuit fabrics. First, it proposes a simple additive error model for timing errors in arithmetic computations. Second, it analyzes the relationship between error statistics and parameters such as the input statistics and the architecture based on the proposed error models. Third, it presents a statistical error characterization methodology based on the proposed error model, thus enabling efficient implementation of emerging stochastic computing techniques. Key results include the following observations: 1) the output error statistics is a strong function of the architecture, and a weak function of input statistics, and 2) the output error statistics depends upon the *one's probability profile* of the input word. These observations enable a one-time off-line statistical error characterization of DSP kernels similar to the delay and power characterizations done presently for standard cells

and IP cores. The proposed error model is derived and verified for a number of DSP kernels in a 45-nm TI CMOS process.

The second part of this chapter addresses engineering of error statistics to enhance the effectiveness of stochastic computing techniques such as soft-NMR and LP. These techniques benefit from not the independence of error events and magnitudes across the redundant processing elements (PEs). For this purpose, architectural diversity and scheduling diversity techniques are proposed to engineer the occurrence of spatially independent error events and magnitudes. Furthermore, the Kullback-Leibler (KL) distance [99] is proposed as an error independence metric to measure the degree of diversity in such systems. The effectiveness of the proposed techniques is demonstrated in the design of a 45-nm soft dual-MR (DMR)-based discrete-cosine transform (DCT) codec. The soft DMR codec achieves a peak signal-to-noise ratio (PSNR) close to that of a triple-MR (TMR) codec even though the former employs one less PEs.

This chapter is organized as follows: Section 6.1 presents the additive error model and its parameters. Section 6.2 analyzes relationships between the output error statistics and the input statistics for a DSP kernel, and presents the error characterization methodology. Section 6.3 verifies the proposed analysis, error model, and characterization methodology for various 45-nm DSP blocks. Architecture and scheduling diversity techniques are presented in section 6.4. Finally, the case study of the DCT image codec is described in Section 6.5.

6.1 Proposed Timing Error Model

We propose that the output of any DSP kernel M' with latched input and outputs (see Fig. 6.1(a)) exhibiting timing errors can be represented via an additive error model (see Fig. 6.1(b)) :

$$y[n] = y_o[n] \oplus e[n] = f_1(x[n], y[n-1], A, V_{dd}, V_t, T, P) \tag{6.1}$$

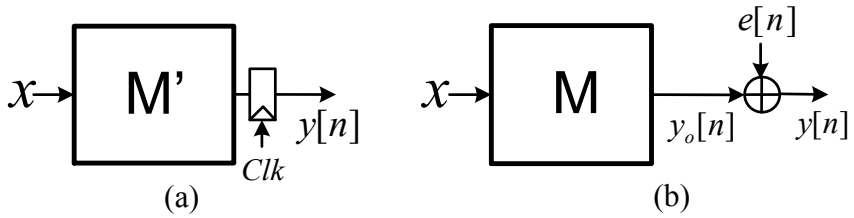


Figure 6.1: A DSP kernel (M') exhibiting errors: (a) block diagram and (b) proposed additive error model.

where $y[n]$ is the corresponding output at time-index (or clock-cycle) n, $x[n]$ is the input, $y_o[n]$ is the correct (error-free) output, and $e[n]$ is the error. For example, if only the second bit in a 3-bit output is in error, then $e = (010)$.

Equation (6.1) indicates that the output $y[n]$ is a complex non-linear function $f()$ of the processing element (PE) architecture (A), the input ($x[n]$), the supply voltage (V_{dd}), the threshold voltage (V_th), the temperature (T), and other physical effects (P). It is also a function of the previous output $y[n-1]$, either because some or all bits of the output $y[n]$ can retain their value if the clock period is too small, or because the architecture is recursive. In the remainder of this chapter, we will focus on non-recursive architectures in order to simplify the exposition and because such architectures can implement a large class of applications.

In non-recursive PEs, $y_o[n]$ is a function of only the present input $x[n]$. Thus, $e[n]$ in (6.1) can be expressed as

$$e[n] = f_2(x[n], y[n-1], A, V_{dd}, V_t, T, P) \tag{6.2}$$

where $e[n]$ embodies all the complex non-linear dependencies on the parameters in the argument of f_2. The dependence of $e[n]$ on $y[n-1]$ reflects the intrinsic memory effects in combinational circuits that are operated faster than the critical path delay. As $y[n-1]$ is

also a function of $x[n-1]$ and $y[n-2]$, we can express $e[n]$ as

$$e[n] = f_3(\mathbf{x}[n], A, V_{dd}, V_t, T, P) \tag{6.3}$$

where $\mathbf{x}[n] = (x[1], x[2], ..., x[n])$. Function $f_3()$ is complex if described in a deterministic manner. Instead, by recognizing that most emerging applications employ statistical performance metrics such as mean-square error (MSE), SNR, and PSNR, and error-aware resilient techniques such as soft NMR that rely on the statistics of $e[n]$ instead of the exact value of error $e[n]$, we propose to treat $e[n]$ as a random variable E and characterize its probability mass function (PMF) denoted by $P_E(k) = p(e[n] = k)$. That is, we are interested in

$$P_E = f_4(P_X, A, V_{dd}, V_t, T, P) \tag{6.4}$$

where P_X represents the PMF of input x. Thus, given a fixed PVT corner, the output error PMF depends on the architectural implementation of the DSP computation and input statistics.

The output error statistics is a strong function of the architecture A or the kernel implementation. Different architectures have different path delay distributions, and thus will result in different errors for the same set of input statistics. In the next section, we study the relationship between the error statistics P_E and input statistics P_X. Specifically, we show that given a DSP architecture A, P_E is relatively insensitive to a large class of input distributions P_X. Thus, the error PMF for each DSP architecure/kernel can be characterized via a one-time off-line procedure at each PVT corner, independent of the application, and similar to power and delay done presently for standard cells. These error PMFs can then be employed by stochastic computing techniques, such as soft NMR and LP, to correct for output errors.

6.2 Error Analysis: Impact of Input Statistics

Many DSP applications have a typical input data set or statistics $P_{X,t}$ which can be employed to characterize the output error of a given architecture. In *communication systems*, knowing the channel noise and transmitted symbol constellation, we can characterize the PMF of the input signals to the receiver kernels. In *media processing systems*, the PMF of pixel values in a given image can represent a large class of images, since adjacent pixels in most images have high data correlations, and thus most image processing kernels have similar input statistics. And in *general work-load DSP*, the input statistics can be assumed to be uniform. Therefore, a typical input data set can be employed to generate the error PMF of a given architecture in a given application. However, this makes error abstraction and characterization procedure dependent on application.

Given a DSP kernel/architecture A, we wish to answer the questions:

1. If we employ an input PMF $P_{X,t}$ to obtain the output error PMF $P_{E,t}$, can we find a class of input PMFs $C_{X,t} = \{P_{X,i}\}_{i=1}^{M}$ such that they all have similar error PMFs as $P_{X,t}$?

2. Can we find a $P_{X,DSP}$ such that the size of the corresponding class, $M = |C_{X,DSP}|$, is large and its characteristics are commonly encountered in most DSP applications?

3. What are the characteristic(s)/condition(s) that the PMFs of $C_{X,t}$ share?

If the answer to the first two questions is in the affirmative, then error characterization can be done once for DSP kernels/architectures employing $P_{X,DSP}$. We show that this is indeed the case. To demonstrate this fact, we study the relationship between input statistics and output error.

As Boolean computation occurs at bit-level, it is expected that the output error statistics P_E will be a stronger function of *bit-level input statistics* rather than *word-level input statistics* P_X. In what follows, 1) we study the relation between word-level and bit-level input

statistics, 2) analyze the impact of bit-level input statistics on output error statistics, and 3) generate a representative input PMF $P_{X,DSP}$ for different DSP applications by conditioning on bit-level statistics of the input PMFs in $C_{X,DSP}$.

6.2.1 Bit-level vs. Word-level Statistics

Any B_x-bit signal/operand $x[n]$ in a DSP kernel consists of bits denoted by $b_{x,i}[n]$ for $i = 1, 2, \ldots, B_x$. We define the following:

- *Bit probability* of $b_{x,i}$: $p_{x,i} = p(b_{x,i}[n] = 1)$
- *Bit probability profile* (BPP) of an operand x: $\Phi_X = (p_{x,1}, p_{x,2}, \ldots, p_{x,B_x})$, i.e., the set of bit probabilities of its constituent bits.
- *Probability mass function* (PMF) of an operand x: $P_X = p(x)$
- $ON(x)$ and $OFF(x)$: bit locations of x whose values are one and zero, respectively, i.e., $ON(x) = \{i | b_{x,i} = 1\}$ and $OFF(x) = \{i | b_{x,i} = 0\}$.

Given a PMF of x, we determine its BPP as follows:

$$p_{x,i} = \sum_{x \in \{x | i \in ON(x)\}} P_X(x) \quad \text{for } i = 1, 2, \ldots, B_x \tag{6.5}$$

i.e., the i^{th} bit probability is obtained by summing P_X over x whose i^{th} bit is one. On the other hand, given a BPP Φ_X, a unique P_X cannot be obtained unless the correlations between bits $b_{x,i}$ are explicitly specified. This is summarized in the following property.

Property 1. *Given any two operands x_1 and x_2:*
$P_{X_1} = P_{X_2} \overset{\Leftrightarrow}{\Rightarrow} \Phi_{X_1} = \Phi_{X_2}$

In fact, the next property shows that the number of P_X that can be mapped to the same Φ_X is very large. Thus, we can define conditions on Φ_X instead of P_X to enforce similar

output error statistics for a given DSP kernel without excessively restricting the input space. In the next section, we study the relation between Φ_X and P_E in order to determine these conditions.

Property 2. *For a fixed precision B_x:*
P_x is symmetric around the mean $\mu_x = \frac{2^{B_x}-1}{2} \Leftrightarrow \Phi_x = (0.5, 0.5, ..., 0.5)$, i.e., $p_{x,i} = 0.5$ for all $i = 1, 2 \ldots, B_x$

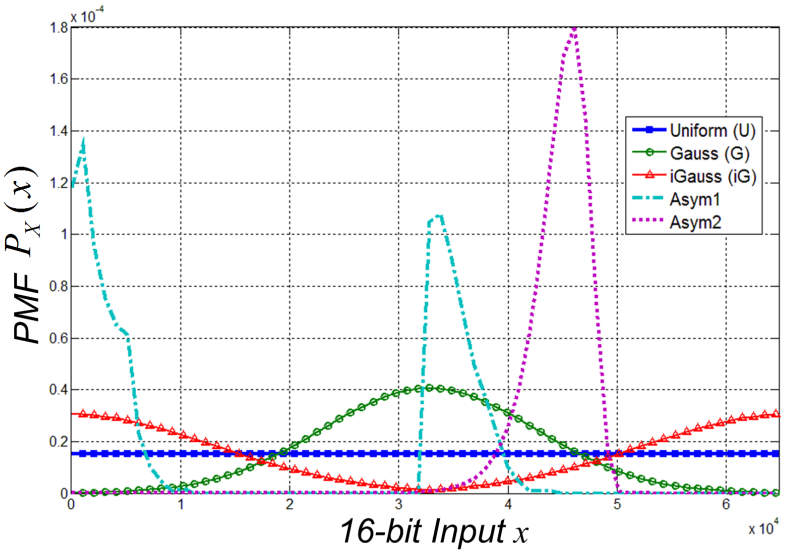
Property 2 indicates that any PMF of x that is symmetric around $\mu_x = \frac{2^{B_x}-1}{2}$ is mapped to the same BPP where each bit is equally likely to be zero or one. Figures 6.2(a) and (b) show a set of different 16-bit input distributions and their respective BPPs. Symmetric distributions (U, G, and iG) with mean $\mu_x = \frac{2^{16}-1}{2}$ have the same equally-likely BPPs where each $p_{x,i} = 0.5$, unlike asymmetric distributions (Asym1 and Asym2).

6.2.2 Impact of Bit-Level Input Statistics on Output Error

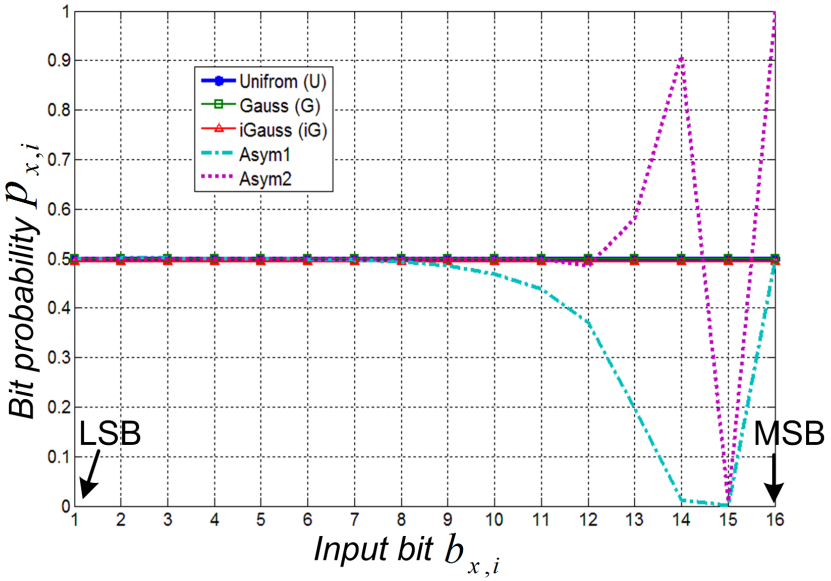
We aim to define conditions on any two different input PMFs, $P_{X,1}$ and $P_{X,2}$, such that the corresponding output error PMFs, $P_{E,1}$ and $P_{E,2}$, are equal. Here, we show that the output error statistics of a given DSP kernel is more dependent on the input BPP, Φ_X, instead of the word-level input PMF, P_X. Thus, condition(s) to ensure similarity of output error statistics can be placed on Φ_X instead of P_X.

Any output signal y_i of a DSP kernel/architecture with input x can be viewed as a cascade of L_i *processing elements* (PEs), denoted by $\{PE_k\}_{k=1}^{L_i}$ (see Fig. 6.3). Each PE_k has an output signal(s) z_k, *intermediate input signal(s)* z_{k-1}, and a *direct input signal set* $x_k \subseteq x$. For example, in case of a carry-ripple adder, the 1-bit full-adders are the PE_k's, the rippled carries are the intermediate signals z_{k-1}, and the k^{th} bits of the input operands are the direct input signal set x_k.

Note that this representation can take place at different granularity levels. For example, each PE_k can represent a single or multiple PEs or even a single logic gate. In what



(a)



(b)

Figure 6.2: Various 16-bit input statistics: (a) word-level distribution and (b) their corresponding bit probability profiles (BPPs).

follows, we decompose the main DSP kernel into PE_k's in such a way that z_{k-1} and x_k are independent. For example, if both z_{k-1} and x_k are generated from the same set of signals, then they are correlated (e.g. $z_{k-1} = a$ and $x_k = \bar{a}$), and in that case we have to enlarge PE_k to make z_{k-1} an internal signal. With such decomposition, if we know the logic functions implemented by all $PE_{j|j\leq k}$, then the probability of any z_k is completely determined by the BPP (Φ) of $x_{j|j\leq k}$ i.e.,

$$p(z_k) = f_k(\Phi_{x_{j|j\leq k}}) \tag{6.6}$$

where $f_k(\cdot)$ is a polynomial function that depends on the logic functions of $PE_{j|j\leq k}$. To see this, for example, if we assume the first PE, PE_1, is a NAND gate with independent 1-bit input signals, x_1 and z_0, and output $z_1 = \overline{x_1 \cdot z_0}$, then $p(z_1 = 0) = p(x_1 = 1)p(z_0 = 1) = p_{x_1}p_{z_0}$ and $p(z_1 = 1) = 1 - p_{x_1}p_{z_0}$. Similarly, one can determine $p(z_2)$ if the logic function of PE_2 is known. This process is continued until PE_k is processed, so that we can express $p(z_k)$ as a polynomial function of $\Phi_{x_{j|j\leq k}}$.

Timing violations occur when the computation for output y_i is not allowed to complete. Assume that each PE_k in Fig. 6.3 PE_k has the same delay d and the clock period is $d(L_i-1)$, i.e., at most $L_i - 1$ PEs can compute correctly. A timing error occurs at the output if all L_i PEs' outputs $z_k[n]$ change their values from the previous clock cycle. If we denote the *transition event* of a signal z_k as t_{z_k}, i.e., $t_{z_k} = 1$ if $z[n] \neq z[n-1]$, then *the probability of output y_i being in error*, $pe_{y,i}$, is expressed as:

$$\begin{aligned} pe_{y,i} &= \sum_{\Phi_X} p(t_{z_1} = 1, t_{z_2} = 1, \ldots, t_{z_{L_i}=1}|\Phi_X)p(\Phi_X) \\ &= \sum_{\Phi_X} \prod_{k=1}^{L_i} \left[p\left(t_{z_k} = 1 | \{t_{z_j} = 1\}_{j=1}^{k-1}, \Phi_X \right) \right] p(\Phi_X) \end{aligned} \tag{6.7}$$

However, the input signal set for each PE_k, denoted by $I_{z_k} = \{z_{k-1}, x_k\}$, shields z_k from signal transitions in preceding PEs, i.e., $p(z_k|I_{z_k}, w) = p(z|I_{z_k})$ where w is any signal in PE_j for $j = 1, 2, \ldots, k-1$. For example, in the case of a ripple-carry adder, given the input carry

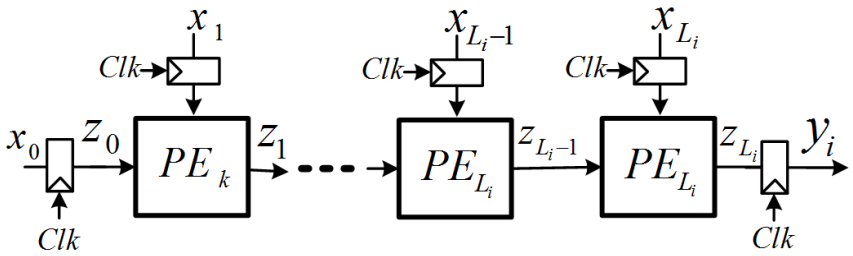


Figure 6.3: An architectural model of a DSP kernel with input x, output bit $b_{y,i}$, and L_i processing elements (PE)s.

and the input bits into the k^{th} 1-bit full-adder, the probability of output carry is independent of signals in the preceding full-adders. Thus,

$$p\left(t_{z_k} = 1 | \{t_{z_{j=1}}\}_{j=1}^{k-1}, \Phi_X\right) = p\left(t_{z_k} = 1 | t_{z_{k-1}} = 1, \Phi_{X_k}\right) \tag{6.8}$$

Substituting in (6.7), we write

$$pe_{y,i} = \sum_{\Phi_X} \prod_{k=1}^{L_i} \left[p\left(t_{z_k} = 1 | t_{z_{k-1}} = 1, \Phi_{X_k}\right)\right] p(\Phi_X) \tag{6.9}$$

In addition, t_{z_k} is relatively independent of $t_{z_{k-1}}$ since z_k is determined by x_k as well, i.e., transitions in z_{k-1} do not necessarily imply transitions in z_k. Thus, (6.10) is expressed as

$$pe_{y,i} = \sum_{\Phi_X} \prod_{k=1}^{L_i} \left[p\left(t_{z_k} = 1 | \Phi_{X_k}\right)\right] p(\Phi_X) \tag{6.10}$$

For ease of notation, we denote $(z_{k-1}[n], z_k[n])$ as $\mathbf{z}[n]$ and introduce the operator $\models$ to denote that all individual components of the two vectors $\mathbf{z}[n]$ and $\mathbf{z}[n-1]$ are not equal. In non-recursive architectures, the signal transitions are independent across time, and the conditional transition probability $p(t_{z_k} = 1 | \Phi_{X_k})$ in (6.10) at the output of each PE_k is

expressed as follows:

$$p(t_{z_k} = 1|\Phi_{X_k}) = \sum_{z_k[n-1] \neq z_k[n]} p(z_k[n-1]|\Phi_{X_k}) p(z_k[n]|\Phi_{X_k}) \tag{6.11}$$

This means that we treat the logic state of PE_k independent of time and sum over values where both $z_k[n]$ and $z_k[n-1]$ are different. For example if z_k is 1-bit then we sum over the duple $(z_k[n], z_k[n-1]) \in \{(0,1),(1,0)\}$.

Since the probabilities are stationary, we treat each $p(\mathbf{z}[n]|\Phi_{X_k})$ and $p(\mathbf{z}[n-1]|\Phi_{X_k})$ similarly. Substituting (6.6) into (6.11) and then (6.10), we get:

$$pe_{y,i} = \sum_{\Phi_X} \prod_{k=1}^{L_i} \sum_{z_k[n-1] \neq z_k[n]} f_{k,n}(\Phi_{x_{j|j\leq k}}) f_{k,n-1}(\Phi_{x_{j|j\leq k}}) p(\Phi_X) \tag{6.12}$$

If we assume that at most $B \leq L_i - 1$ PEs compute correctly, then, for an error to appear at the output, all PE_k for $Q_i = L_i - B - 1 \leq k \leq L_i$ need to undergo a transition in clock-cycle n, i.e., the last B PEs need to undergo a transition independent of preceding PEs in the chain. Otherwise the error cannot be propagated. Conditioning on $p(z_{Q_i-1})$ will shield all $PE_{k>Q_i-1}$ from signal transitions in preceding PEs in the logic chain, and thus (6.9) is written as:

$$\begin{aligned} pe_{y,i} &= \sum_{\Phi_X} p(\Phi_X) \sum_{z_{Q_i-1}} p(z_{Q_i-1}) \\ &\quad \prod_{k=Q_i}^{L_i} \left[p\left(t_{z_k} = 1 | t_{z_{k-1}=1}, \Phi_{X_k}, z_{Q_i-1}\right) \right] \end{aligned} \tag{6.13}$$

Following similar procedure from (6.9) to (6.12), $pe_{y,i}$ in (6.13) can also be written as a polynomial function of Φ_X.

This shows that knowing Φ_X completely determines the probability of output errors in a given DSP kernel. Thus, we can modulate the probability of output error of a DSP block by enforcing conditions on the constituent elements of Φ_X. Next, we employ this observation

to generalize the proposed error model to be independent of the application, given a DSP architecture.

6.2.3 Generalized Error Characterization

Given a DSP kernel/architecture A and two input statistics $P_{X,1}$ and $P_{X,2}$ that have the same BPP, i.e., $\Phi_{X,1} = \Phi_{X,1} = \Phi_X$, then (6.13) shows that output error PMFs corresponding to the two input PMFs are equal, i.e., $P_{E,1} = P_{E,2} = P_E$. Moreover, Property 2 shows that for a DSP kernel with input precision B_x, all input PMFs that are symmetric around $\frac{2^{B_x}-1}{2}$ have a BPP where all bits are equally likely. We denote this BPP as $\Phi_{X,U}$ and define the corresponding class of PMFs as $C_{X,U}$. The uniform input distribution U can be used as a representative input distribution to characterize the DSP kernel for $C_{X,U}$. Moreover, the class of input PMFs $C_{X,U}$ can be generalized further to include any input PMF that is symmetric around any value $\mu_x \in (0 : 2^{B_x-1})$. We denote this class as $C_{X,DSP}$ and the uniform input distribution U can still be used as a representative for error characterization to obtain $P_{E,DSP}$ of $C_{X,DSP}$. To see this, the mean of $x' = x + \frac{2^{B_x}-1}{2} - \mu_x$ is $\mu_{x'} = \frac{2^{B_x}-1}{2}$ and thus $P_{X'} \in C_{X,U}$ and the corresponding error PMF of x' is $P_{E'} = P_{E,U}$. Then, the error PMF of x can be obtained from the error-free DSP kernel functionality f_{DSP} via a simple translation of $P_{E,U}$ as follows:

$$P_E = P_{E,U} + f_{DSP}\left(\mu_x - \frac{2^{B_x}-1}{2}\right) \tag{6.14}$$

Therefore, output error characterization for a DSP kernel/architecture at a given PVT corner can be done once using a uniform input distribution to obtain $P_{E,DSP}$. The obtained error PMFs $P_{E,DSP}$ of the DSP kernel/architecture is applicable to any application whose input statistics is symmetric. A large class of DSP applications can thus use $P_{E,DSP}$. For example, any application whose input distribution is uniform, Gaussian, or iGaussian as shown on Fig. 6.2(a) or is a mixture of any set of symmetric distributions will have the same

output error PMF $P_{E,DSP}$. If the input statistics in a given application $P_{X,as}$ was found out to be asymmetric, then the error-characterization needs to be redone for the DSP kernel taking into account $P_{X,as}$.

Given a DSP kernel/architecture A, an operating frequency f_{op}, and synthesis libraries at different PVT corners, the generalized error characterization flow is as follows:

1. Generate a uniformly distributed input data set $D_{x,U}$ and obtain the corresponding error-free output $y_o[n]$ using an RTL or fixed-point simulation of the DSP kernel.

2. Synthesize the design at a PVT corner to obtain a gate-level netlist of the DSP kernel that can operate timing error-free at f_{op}.

3. Back-annotate the synthesized gate-level netlist with timing information (standard delay format (SDF) file) at PVT corners worse than the synthesis PVT corner in step 2, i.e., at supply voltages lower than the synthesis voltage and/or process corners slower than the synthesis process corner.

4. Generate the erroneous output $y[n]$ at different PVT corners by employing an RTL-level simulation of the synthesized gate-level netlist in step 2 using the same input data set $D_{x,U}$ as step 1 and the SDF files generated in step 3 while fixing the operating frequency at f_{op}.

5. Error PMF P_E is obtained at different PVT corners by comparing $y_o[n]$ in step 1 to $y[n]$ in step 3 according to the relation in (6.1).

6.3 Simulations and Verifications for Statistical Error Characterization

To validate the error analysis, modeling, and characterization, we employ *voltage overscaling* (VOS) in order to generate timing violations and thereby emulate PVT variations. In VOS,

the supply voltage is reduced below a critical supply voltage $V_{dd-crit}$, which is the lowest voltage at which the system operates error-free, while keeping the frequency of operation fixed at f_{op}. Thus, intermittent timing errors $e[n]$ will appear at the output. We define $V_{dd}/V_{dd-crit}$ as the voltage overscaling factor K_{VOS}. In what follows a 45-nm TI CMOS process is employed and error PMF P_E of a given DSP kernel/architecture A is obtained at each voltage following the characterization flow outlined in the previous section. In certain cases, when we want to study the effect of different input statistics on output error, we use the respective input statistics instead of a uniform one to perform error characterization of the DSP kernel. We focus on adder and multiplier units since these are widely used in DSP designs and form most of the data path in circuits benchmarks such as ISCA-85/89.

We employ *Kullback-Leibler* (KL) distance [99] to quantify the difference between error PMFs for different input statistics and architectures. Given two PMFs P_{E_1} and P_{E_2} of two random variables E_1 and E_2, the KL-distance is:

$$KL(P_{E_1}, P_{E_2}) = \sum_{e} P_{E_1}(e) \log_2 \frac{P_{E_1}(e)}{P_{E_2}(e)} \tag{6.15}$$

KL distance measures the distance between two distributions so that $KL(P_{E_1}, P_{E_1}) = 0$ if and only if $P_{E_1} = P_{E_1}$. Usually, two PMFs are quite similar if KL distance < 1.

6.3.1 Impact of Architecture

The architectural or implementation choice of a DSP function strongly affects the output error behavior since different architectures have different logic paths between input and output. To verify this, we show how error PMF, P_E, varies at the output of different architectures implementing the same DSP functionality with the same input x and error-free output y_o.

We characterize the error PMFs due to VOS using the same uniformly distributed input data-set for three 16-bit adders employing different architectures (ripple-carry adder (RCA),

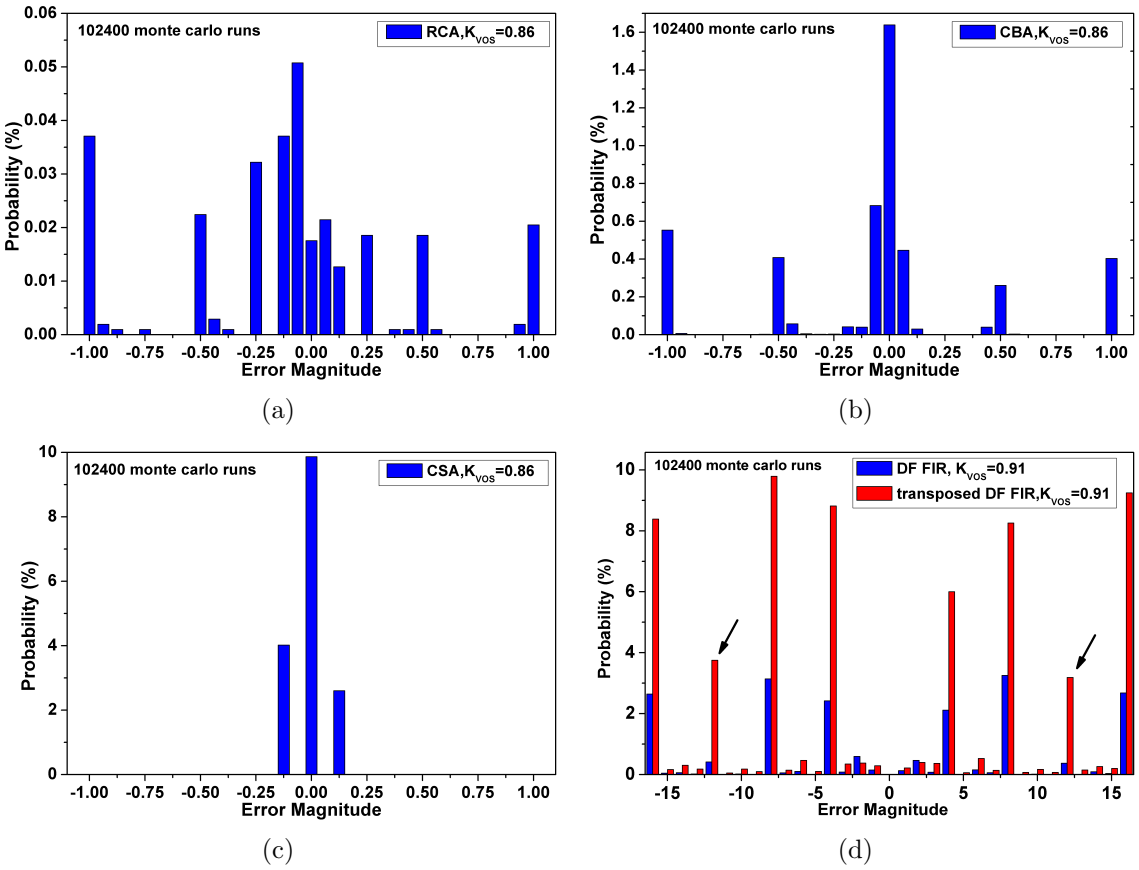


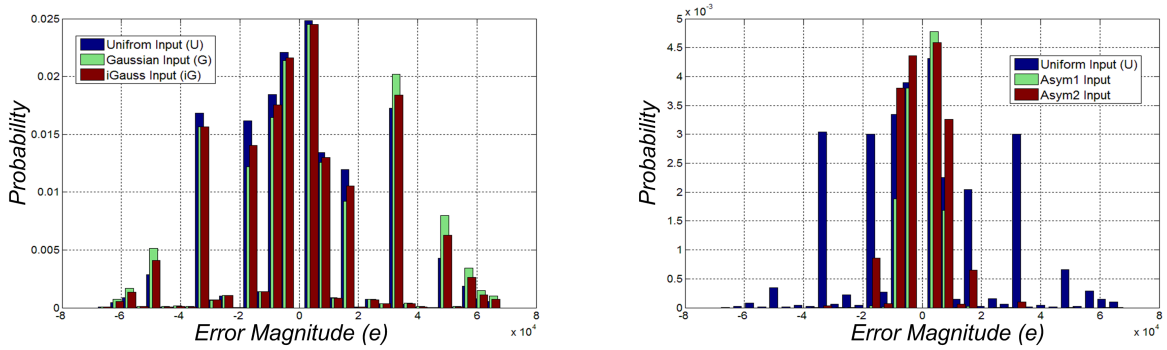
Figure 6.4: Error statistics of various architectures: (a)16-b RCA, (b) 16-b CBA, (c) 16-b CSA, and (d) DF and TDF 16-tap FIR filter.

carry-bypass adder (CBA), and carry-select adder (CSA)). We do the same for an FIR filter with direct-form (DF) and transposed direct-form (TDF) implementation. The FIR filters are 8-bit input 16-tap low-pass filters implemented using Baugh-Wooley multipliers and RCAs. Figure 6.4 shows P_E for the three adder types and for the two FIR filters. The plots in Figs. 6.4(a), (b), and (c) indicate that the three different adders have clearly distinct error PMFs at the same K_{vos}. Similarly in Fig. 6.4(d), the direct and transposed form FIR filters do have distinct error statistics though they have the same input. These conclusions support those of [100] which show that different arithmetic unit architectures have different average error magnitudes. Therefore, error statistics are indeed strongly dependent on the architecture.

Table 6.1 shows that the KL distance is large among the three adder architectures and

Table 6.1: KL-distance between error PMFs in various architectures at different K_{VOS}.

	16-bit Adder			16-tap FIR
K_{VOS}	$KL_{RC,CB}$	$KL_{RC,CS}$	$KL_{CB,CS}$	$KL_{DF,TDF}$
0.95	7.3	9.0	0.4	3.2
0.90	18.3	19.3	11.0	7.1
0.82	26	64	92	15
0.73	69	148	190	32



(a) (b)

Figure 6.5: Output error statistics of 16-bit RCA at $K_{vos} = 0.73$ using: (a) symmetric input statistics P_X's and (b) uniform input distribution and asymmetric P_X's

between DF and TDF FIR filters, indicating that the architecture choice indeed strongly affects error PMF. Note that as voltage is reduced, the KL distance between any two error PMFs increases since more architecturally-different paths become critical and more distinct errors appear at the output of a DSP block.

6.3.2 Impact of Input Statistics

To verify the analysis and the relation between word-level (PMF) and bit-level (BPP) error statistics and output statistics, we use the probability distributions in Fig. 6.2(a) as input statistics for a 16-bit ripple-carry adder (RCA). Figure 6.5 shows the output error PMF of a 16-bit RCA when subject to the different input statistics (U, G, iG, Asym1, and Asym2) at $K_{VOS} = 0.73$. Note that the input PMFs U, G, and iG are symmetric around $\mu = \frac{2^{16}-1}{2}$ and have the same equally likely BPP in Fig. 6.2(b) unlike the other two (Asym1 and

Table 6.2: KL distance between error PMFs of 16-bit adders under various input statistics and error PMF P_{E_U} obtained using a uniform input distribution.

K_{VOS}	KL_{E_U,E_G}	$KL_{E_U,E_{iG}}$	$KL_{E_U,E_{Asym1}}$	$KL_{E_U,E_{Asym2}}$
	16-bit RCA			
0.95	0	0	0.062	0.04
0.90	0	0	0.15	0.06
0.82	0.01	0.01	1.15	0.20
0.73	0.07	0.07	8.86	1.33
0.65	0.30	0.28	52.0	8.48
	16-bit CBA			
0.95	0	0	0.08	0.05
0.90	0	0	3.93	0.06
0.82	0	0	24.3	0.72
0.73	0.02	0.01	32.6	1.83
0.65	0.01	0	142	14.5
	16-bit CSA			
0.95	0	0	0.07	0.07
0.90	0	0	1.29	0.53
0.82	0	0	40.7	0.40
0.73	0.01	0	129	15.7
0.65	0.1	0.02	308	96.5

Asym2). Figure 6.5(a) shows that the output error statistics for the symmetric distributions is quite similar. On the other hand, the output error PMFs in Fig. 6.5(b) corresponding to asymmetric input PMFs are very different from those obtained using the uniform input distribution (U).

Table 6.2 shows the KL distance between the error PMFs corresponding to different input PMFs and the error PMF P_{E_U} obtained using a uniform input distribution in different 16-bit adders. The error PMFs corresponding to symmetric input PMFs, G and iG, have a very small KL distance with P_{E_U}. On the other hand, error PMFs corresponding to asymmetric input PMFs Asym1 and Asym2 are close to P_{E_U} only at high K_{VOS} where the voltage of the adder is not reduced enough to produce a large number of output errors. As voltage is reduced further, the error PMF of asymmetric input distributions starts to have a very large KL distance compared to P_{E_U}. Note that $KL(P_{E_U}, P_{E_{Asym1}})$ is greater than $KL(P_{E_U}, P_{E_{Asym2}})$ indicating that, when compared to P_{E_U}, the error PMF due to Asym1

Table 6.3: KL distance between error PMFs of a 16-tap FIR filter under various input statistics and error PMF P_{E_U} obtained using a uniform input distribution.

K_{VOS}	KL_{E_U,E_G}	$KL_{E_U,E_{iG}}$	$KL_{E_U,E_{Asym1}}$	$KL_{E_U,E_{Asym2}}$
	Direct-Form FIR			
0.95	0.06	0.04	21.6	0.05
0.90	0.94	0.15	63	3.57
0.82	0.92	0.14	33	3.10
0.73	0.03	0.82	227	209
	Transposed-Form FIR			
0.95	0.49	0.13	70	0.53
0.90	0.91	0.38	62	5.78
0.82	0.31	0.08	56	3.41
0.73	0.03	0.89	203	163

is much different than that of Asym2 because the Asym1 PMF is more asymmetric than Asym2 PMF (see Fig. 6.2(b)). A similar trend is observed in Table 6.3 for different types of 16-tap FIR filters where error PMFs of symmetric input distributions are close to P_{E_U} while those of asymmetric distributions are quite different. These results support the presented error analysis and modeling procedure and, specifically, the fact that input distributions with similar input BPPs produce similar output error statistics.

6.4 Diversity Techniques for Error Independence

Conventional NMR requires that the error events across the replicated modules be independent to avoid common-mode failures (CMFs), i.e., correlated error events, so that the majority voter would not fail catastrophically. Also, a large class of fault-tolerant systems are based on DMR with re-computation when an error is detected. These systems require that the replicated modules produce non-identical error values/magnitudes to avoid undetectable errors. The authors in [77] introduced the D-metric to measure the diversity degree of such systems, i.e., the probability of producing non-identical errors across two modules

given by:

$$D = \sum_{\{(e_1,e_2):e_1 \neq e_2\}} p\left(e_1, e_2 | \textit{ an error occured}\right) \tag{6.16}$$

where e_1 and e_2 are the error values appearing at the output of two modules. Thus, the higher the D-metric is the more reliable the NMR system is.

Emerging stochastic computing techniques such as soft-NMR and the proposed LP technique can operate effectively in the presence of identical error values, but benefit from having the error magnitudes – not only the error events – be independent across the modules in order to reduce the complexity overhead and improve robustness to errors. Thus, diversity techniques [101], [102], and diversity metrics for conventional NMR, such as the D-metric, are no longer relevant in emerging robust system-design techniques. In this part, we propose *architectural diversity* and *scheduling diversity* to support these emerging techniques. We show that *architectural* and *scheduling* diversity techniques are simple to apply and yet highly effective in ensuring that errors are spatially independent. Furthermore, we employ the KL distance defined in (6.15) to measure the degree of independence between two random variables E_1 and E_2 representing the errors at the output of two PEs, and consequently measuring the efficiency of the diversity techniques. Finally, as a practical demonstration, we show how error characterization and engineering in a DCT-based image codec leads to enhanced robustness and energy efficiency.

6.4.1 Architectural Diversity

The architectural or implementation choice of a DSP function strongly affects the output error behavior since different architectures have different logic paths between input and output [95] [100]. This makes *architectural diversity* an attractive candidate to generate independent error magnitudes in NMR even though all modules have the same input. We employ architectural diversity in the design of a dual-modular redundant (DMR) 16-bit adder and 16-tap FIR filter, and measure the independence of output errors. *Voltage overscaling*

Table 6.4: Error independence between RCA, CBA, and CSA, where $V_{dd-crit,RCA} = 1.1\,\text{V}$, $V_{dd-crit,CBA} = 0.95\,\text{V}$, $V_{dd-crit,CSA} = 0.85\,\text{V}$, and $f = 1.01\,\text{GHz}$

K_{VOS}	$p_{CMF}(\%)$	$D(\%)$	KL_{E_1,E_2}
	RCA and CBA		
0.95	0.0	100	0.001
0.90	0.0	100	0.004
0.85	0.3	99.999	0.025
	RCA and CSA		
0.95	0.0	100	0.001
0.90	0.1	100	0.002
0.85	0.3	99.998	0.009
	CBA and CSA		
0.95	0.0	100	0.001
0.90	1.6	99.987	0.036
0.85	6.7	99.976	0.154

(VOS) is employed in the 45-nm TI CMOS process in order to generate timing violations and thereby emulate PVT variations in computation. In VOS, the supply voltage is reduced below a critical supply voltage $V_{dd-crit}$, which is the lowest voltage at which the system operates error-free, while keeping the frequency of operation fixed at f_{op}. Thus, intermittent timing errors $e[n]$ will appear at the output. We define the ratio $V_{dd}/V_{dd-crit}$ as the voltage overscaling factor K_{VOS}. The different architectures are operating at the same clock frequency but each having its own $V_{dd-crit}$ in order to meet the timing constraints imposed by the system clock. To observe the output error behavior across the different architecture simultaneously, gate-level simulations are carried out at each supply voltages while the same input is fed to all architectures. A total of 10^7 input vectors, sampled from a uniform distribution, are used.

Three architectural candidates are considered for the 16-bit adder: RCA, CBA, and CSA. Table 6.4 quantifies the dependence of errors (E_1, E_2) at the outputs of a pair of adders employing different adder architectures using KL distance and shows conventional measures such as D-metric and p_{CMF}, the probability of common mode failures, i.e., error events where a conventional DMR system fails to detect an error. The output error magnitudes

Table 6.5: Error independence between DF and TDF FIR filters, where $V_{dd-crit,DF} = 1.1\,\mathrm{V}$,$V_{dd-crit,TDF} = 1\,\mathrm{V}$, and $f = 588\,\mathrm{MHz}$.

	DF FIR and Transposed DF FIR		
K_{VOS}	$p_{CMF}(\%)$	$D(\%)$	KL_{E_1,E_2}
0.95	1.1	99.628	0.007
0.90	16.2	97.952	0.029

are almost independent for any pair, especially for the RCA-CSA pair, which has the lowest $KL(P_{E_1,E_2}, P_{E_1}P_{E_2})$ making it the best choice for advanced robust system design techniques. This conclusion contrasts with that obtained using conventional measures (p_{CMF} and D-metric in Table 6.4) which indicate that the RCA-CBA pair is better for conventional NMR since it has the smallest probability of CMFs.

For the 16-tap FIR filter, two architectures are considered: direct-form and transposed direct-form. The FIR filters are 8-bit input 16-tap low-pass filters implemented using Baugh-Wooley multipliers and RCAs. Table 6.5 shows that errors are indeed independent between the two architectures. Therefore, *architectural diversity* is an effective way to make error magnitudes independent in advanced error-resilient designs.

6.4.2 Scheduling Diversity

Besides *architectural diversity*, we propose another general and simple diversity technique, *scheduling diversity*, to make errors independent across redundant outputs. To reduce hardware complexity and increase power efficiency, folding is a well-known technique which executes similar operations on the same hardware. In *scheduling diversity*, we reorder the sequence of operations to be executed on multiplexed modules. A 15-tap FIR filter is employed as an example to demonstrate scheduling diversity, whose output at time n is given by

$$y[n] = \sum_{k=0}^{14} h[k]x[n-k] \tag{6.17}$$

Table 6.6: Error independence with scheduling diversity, where $V_{dd-crit} = 1.1\,\text{V}$ and $f = 714\,\text{MHz}$.

	Schedule 1 and 2		
K_{VOS}	$p_{CMF}(\%)$	$D(\%)$	KL_{E_1,E_2}
0.95	7.0	96.425	0.027
0.90	26.6	96.574	0.119
	Schedule 1 and 3		
0.95	8.9	95.548	0.051
0.90	29.5	96.624	0.137
	Schedule 2 and 3		
0.95	8.9	95.114	0.011
0.90	26.1	97.550	0.091

where $h[k]$ is the filter coefficient, and $x[k]$ is the input. Different schedules can be employed to map (6.17) onto a single multiply-accumulate (MAC) unit. We employ three possible schedules, and measure the independence of errors in Table 6.6. We observe that output errors for the three schedules are pairwise independent even if p_{CMF} is high.

6.5 Case Study: Discrete-Cosine Transform (DCT) Codec Design

We demonstrate the use of error statistics and diversity techniques in the design of a robust DMR-based DCT codec shown in Fig. 6.6(a). The DCT and inverse-DCT (IDCT) transform are applied to a 256×256 8-bit pixel image in blocks of 8×8 pixels using Chen's algorithm [97]. Each two-dimensional (2D) transform is implemented by applying a 1D transform row-wise, and then column-wise on the output of the first. Transposition memory (TM) is used to swap the data between rows and columns. The quantizer (Q) and inverse quantizer (Q^{-1}) employ the JPEG quantization table. The error-free codec achieves a peak signal-to-noise (PSNR) ratio of 33 dB.

In DMR, two codecs are employed so that the two outputs y_1 and y_2 are available to the voter. To ensure error independence between the two codecs, *scheduling diversity* is applied by swapping the inputs of the array-based multipliers in the redundant codec. We obtain

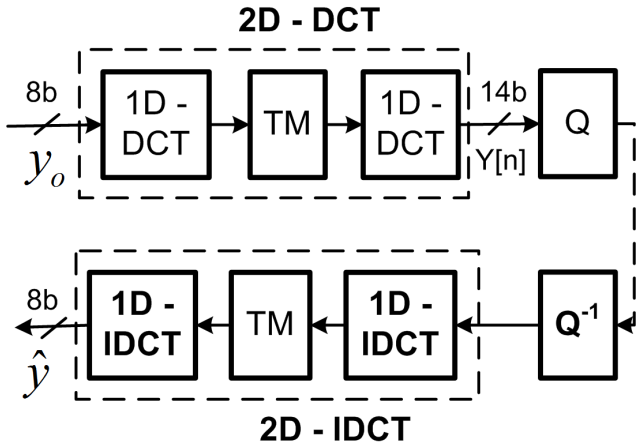


Figure 6.6: Block diagram of the 2D DCT-IDCT codec.

Table 6.7: Error independence of two voltage overscaled DCT codec using different scheduling.

K_{VOS}	$p_{CMF}(\%)$	$D(\%)$	KL_{E_1,E_2}
0.96	0	100	0.000
0.92	0.03	100	0.003
0.88	0.05	99.991	0.005
0.83	0.44	99.966	0.040
0.79	4.29	99.835	0.257
0.75	15.09	99.515	0.639
0.71	33.87	98.764	1.300
0.66	58.76	97.531	1.950

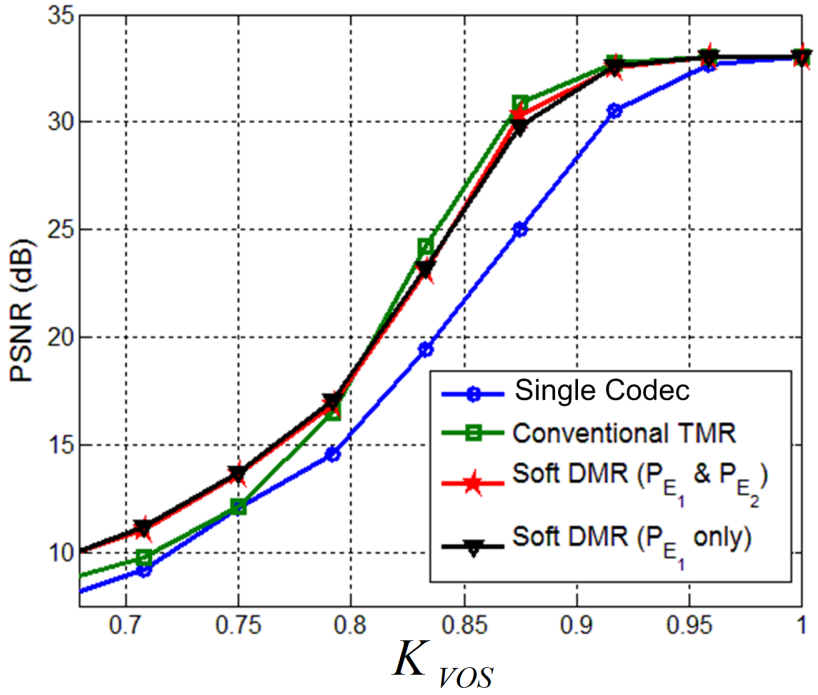


Figure 6.7: Performance of soft DMR-based codec under VOS.

the VOS error statistics P_{E_1} and P_{E_2} at the output of the two codecs respectively. Table 6.7 shows that errors at the output of the two codecs are indeed independent. Note that, as VOS increases (smaller K_{vos}), the error dependence measure (KL distance) increases since the number of erroneous paths for two codecs increases especially at lower V_{dd}. However, the KL distance is still small even at low V_{dd} (maximum value of KL distance is 8 for the 8-bit output codec), and thus, the errors are still independent.

Conventional DMR can only detect errors and relies on re-computation to correct for them, while soft DMR [78] utilizes error statistics to detect and correct errors. Using a look-up table that stores the pre-characterized error statistics P_{E_1} and P_{E_2} of the two codecs, the soft voter employs the *maximum likelihood*(ML) rule at the outputs of the two voltage-overscaled RTL codecs to select the output with the higher probability of occurrence (see [78] for a low-overhead implementation of the soft voter). Figure 6.7 shows the robustness of soft DMR at different K_{VOS}. Soft DMR performs close to conventional TMR though it uses

one less codec. Also, the robustness of soft DMR is barely affected in Fig. 6.7 when using a single-error distribution for the two redundant codec, i.e., assuming $P_{E_1} = P_{E_2}$ since they have similar architectures but different scheduling. Note: as K_{VOS} decreases below 0.8, soft DMR starts to perform even better than conventional TMR since p_{CMF} is high (4% to 60% in Table 6.7) and TMR ignores error statistic.

6.6 Summary

We proposed a statistical additive error model that captures the statistical distribution of timing errors in arithmetic units and DSP blocks at architecture and system level. We showed that this model is relatively independent of input statistics for a wide class of applications. Moreover, we presented techniques to ensure error magnitude/value independence across redundant observations (spatial independence) in emerging error-resilient techniques.

CHAPTER 7

CONCLUSIONS

The paradigm shift toward a ubiquitous computing world is characterized by a profusion in embedded ULP platforms where energy and size are of utmost concern for seamless integration and long battery-life. This dissertation the statistical nature of the ULP application performance metrics and the dynamic nature of ULP workload characteristics, and matches them to the statistical attributes and the device region of operation of the underlying circuit/device fabric. This is done while taking into consideration the energy-delivery overhead, resulting in systems that operate closer to the limits of energy-performance envelop.

7.1 Dissertation Contributions

Research on MEOP has primarily been a circuit-level inquiry and is characterized by significant loss in design margins since PVT variations at the MEOP are significant and error resiliency at the MEOP is yet unexplored. Stochastic computing and other error compensation techniques have been applied in the high-throughput superthreshold regime. This dissertation studies the application of stochastic computing at the MEOP. It shows that stochastic computing achieves 28% to 54% energy savings beyond what is achievable by conventional MEOP designs. It demonstrates acceptable application-level performance metrics in the presence of 70% to 85% pre-correction error rate, which represents a 700× to 850× increase in error handling capability as compared to existing error-compensation techniques. These conclusions are further verified by designing a subthreshold stochastic computing-based ECG processor IC in a 45-nm CMOS process. The prototype IC delivers acceptable

beat-detection rates while operating at 15% below its critical supply voltage in the presence of 58% error rate. These results represent an improvement of 19× in beat-detection accuracy, 600× in p_η, and 28% in energy over conventional (error-free) MEOP systems. The prototype IC consumes 14.5 fJ/cycle/1k-gate and exhibits 4.7× better energy efficiency than the state-of-the-art while tolerating 16× more voltage variations.

Stochastic computing allows ULP next-generation applications to continue reaping the energy and size benefits of Moore's law despite the increasingly statistical device behavior. However, low supply voltage operation due to technology scaling and subthreshold operations reduces the efficiency of energy-delivery subsystem. This dissertation enables the design of energy-efficient integrated ULP platforms/systems by jointly optimizing over the core and DC-DC converter design spaces. Significant energy savings (45.5%) are demonstrated by operating at the MEOP of the system (S-MEOP), as compared to the current practice of operating at the core MEOP (C-MEOP). Architectural techniques are proposed to mitigate the energy-delivery overhead at S-MEOP resulting in a 2.3× improvement in system energy efficiency and S-MEOP approaching C-MEOP to within 5%.

This dissertation proposes a novel stochastic-computing technique, which generates reliability information or confidence level on each output bit, and is referred to as *likelihood processing* (LP). The robustness and energy benefits of LP are demonstrated in the design of a 45-nm 2D-DCT codec, which can be employed as a hardware accelerator in a ULP platform. Results show 5× to 100× improvement in robustness along with 15% to 71% energy savings compared to conventional (error-free) and existing stochastic computing techniques.

It is clear that the availability of statistical hardware error models, and developing an understanding of the factors that impact these models, are essential in the investigation and development of robust stochastic computing design principles. This dissertation proposes a unified framework for stochastic computing paradigm, develops a statistical error-model for robust DSP-heavy computations, and shows that the proposed model is effective in abstracting the hardware-error behavior at system level. The different factors affecting error

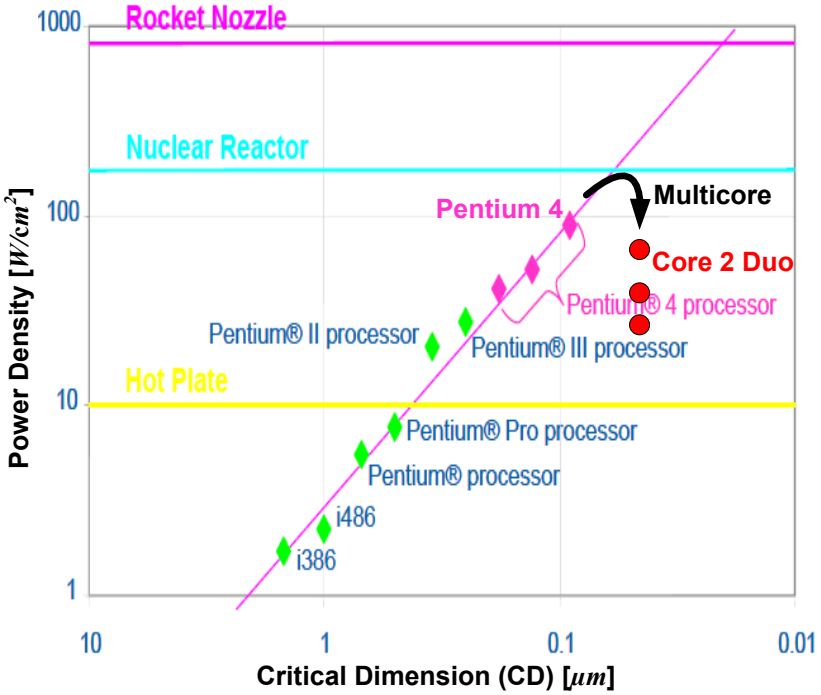


Figure 7.1: The power wall in CPU design [103].

statistics under the proposed model are studied, and a one-time off-line error characterization methodology is proposed, which is similar to the power and delay characterization done today for standard cells and IP cores. Finally, design diversity techniques are proposed to engineer favorable spatially-independent error statistics.

7.2 The Broader Impact: Beyond ULP Platforms

This dissertation has addressed energy efficiency and robustness in ULP platforms, enabling them to operate dramatically closer to the limits of the achievable robustness-energy-performance envelope. This dissertation is distinguished by its integration of principles from power electronics, statistical signal processing, estimation and detection, VLSI architectures, and IC design. The stochastic design philosophy and principles introduced in

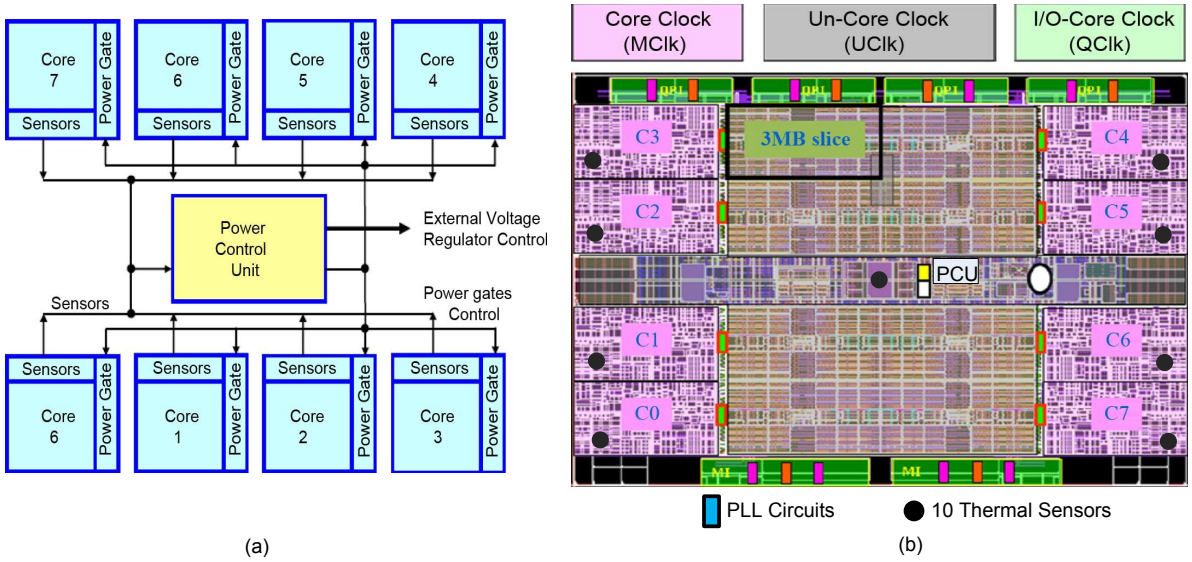


Figure 7.2: The 45-nm 8-core *Intel Enterprise Xeon* processor: (a) block diagram with on-chip power management and (b) die photo with multiple clock domains and thermal sensors [104].

this dissertation and their demonstrated robustness and energy benefits can be extended to next-generation high-throughput processors and platforms.

In the last three decades, performance (speed) had been the main central processing unit (CPU) design metric, with technology scaling as its workhorse. Device scaling has led to faster circuits, smaller silicon area, and reduced supply voltages. Thus, in spite the CPU speed and functional complexity increase, its power density (power per unit area) remained almost constant for the same chip area in old process technologies. However, in sub-micrometer scale processes, the reduction in the supply voltage has been limited by the threshold voltage, leading to an increase in power density. For example, a 3 GHz *Pentium 4* CPU has a power density close to a nuclear reactor (see Fig. 7.1). This problem was aggravated further by microarchitectural design techniques such as instruction-level parallelism, out-of-order execution, and multi-threading, which focused solely on throughput enhancement. Hitting the power wall (the chip's overall power budget due to cooling constraints) in single-core designs has forced industry to shift recently towards heterogeneous multi- and

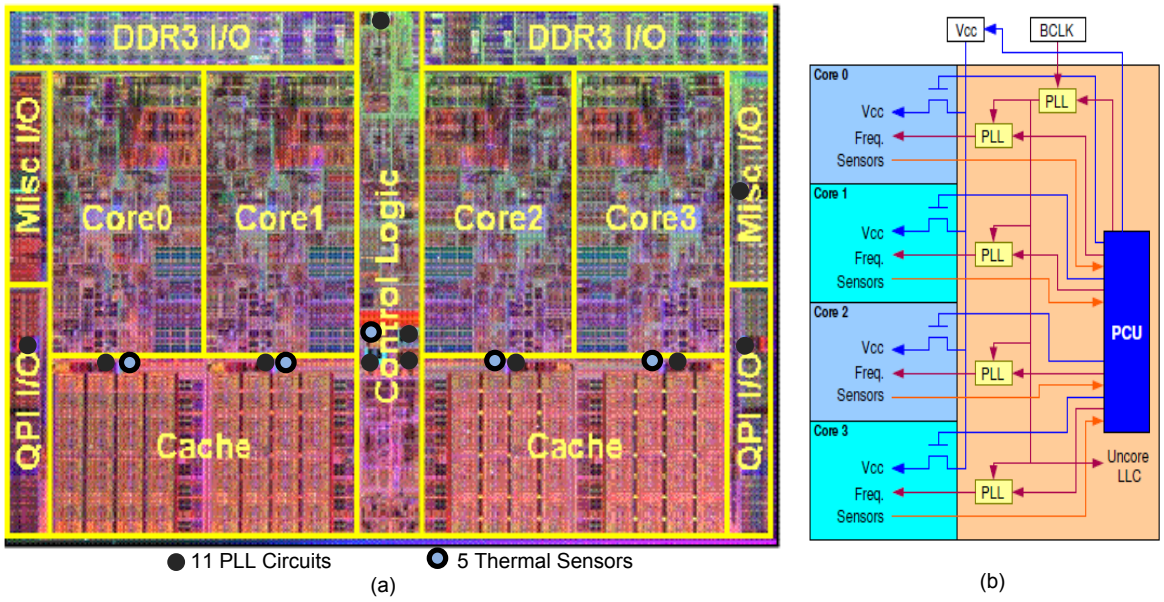


Figure 7.3: The 45-nm *Intel Core i7* (quad core) processor: (a) die photo and (b) on-chip power management to enable DVS and power gating [105].

many-core architectures (see Fig. 7.1) where explicit application- and thread-level parallelism is exploited in a power efficient manner. In fact, modern CPUs are being designed as complex SoCs with massive interfaces, multiple diverse and specialized functional units, which are power gated and dynamically voltage scaled depending on workload characteristics (see Figs. 7.2 and 7.3) [104–107]. Modern processors are converging toward the SoC-based architecture, similar to those found in ULP platforms. Thus, system-design principles proposed in this dissertation can be similarly applied to modern processors to aid them in overcoming the power wall and coping with the increasingly unreliable device/circuit fabric. For example a stochastic-based approach to modern processor design will enable VOS in modern processors to considerably save power beyond conventional techniques while maintaining the application performance metric and the processor speed requirements.

7.3 Future Work

Stochastic computing for ULP platforms introduces a paradigm shift in systems and hardware design with tremendous increase in robustness and energy savings. Thus, extending the applicability and benefits of stochastic computing across all levels of modern system design in silicon and post-silicon technologies will have significant impact on future systems and platforms. This opens up a number of interesting problems to further explore. Although these problems are interlinked, they can be characterized across five domains: 1) application and software, 2) systems, 3) architecture and CAD support, 4) circuits and devices, and 5) theoretical foundations. A synergetic research effort is needed, which requires seeking collaboration with researchers in various research areas such as nanotechnology, computer architecture, communications, digital signal processing, machine learning, and computer vision.

7.3.1 Applications and Software Domain

It is fortuitous that next-generation applications depend heavily on sensing surveillance and media-rich immersive computing [1]. The recognition and mining aspects of such applications make them heavily dependent on advanced signal processing and classification (machine learning) mechanisms, such as support vector machines and Bayesian classifiers, which achieve high probabilities of abnormal event detection. The increased power and complexity of these error-tolerant kernels provide a good opportunity for applications of stochastic computing techniques.

Exposing the architectural-level hooks to the software would provide additional benefits over the current practice where stochastic computing has been applied in hardware. For example, Chapter 5 showed how soft information can be generated by LP. This information can be exploited at the software level to enable optimal task allocation and power management policies. Furthermore, Chapter 4 investigated core error-resiliency assuming a worst-case

scenario of voltage droop/ripple. Incorporating task scheduling and software profiling techniques will improve the implementation strategy of a jointly designed stochastic core and DC-DC converter. Indeed, [108] shows that task scheduling can be employed in multicore platforms to mitigate the effect of voltage droop and achieve better energy efficiency.

7.3.2 Systems

Advanced processing of hardware error statistics increases the energy and robustness benefits of stochastic computing. To this end, advanced statistical techniques from machine learning and communications, such as belief propagation and approximately decodable codes [109], present good candidates to exploit error statistics more efficiently. An interesting problem would be to develop iterative/turbo versions of LP in Chapter 5 where different likelihood processors exchange their soft information to achieve increased energy efficiency and robustness.

7.3.3 Architecture and CAD Support

Stochastic computing has been applied to DSP-heavy applications where statistical performance metrics are employed. Architecture-level research is required to overcome the application specificity of stochastic computing and extend its deployment to general-purpose architectures. Architectural techniques are further necessary to engineer associated error statistics into forms that are amenable to stochastic computing techniques, e.g., see Chapter 6 where diversity techniques are introduced to achieve spatial architecture-level error independence.

Composability of the error model is a desired property for statistical error characterization. Composable error models enable us to build architectural error models from those of their constituents. Statistical error characterization and engineering of architecture and circuit

macros needs to be done along with power and delay modeling, in order to enable the design of robust energy-efficient systems.

7.3.4 Circuits and Devices

Circuit-level techniques need to be investigated in order to support architecture techniques in generating favorable error statistics as well as voltage-ripple-tolerant designs. A variety of research has also emerged on post-silicon technologies such as carbon nanotubes, phase change memory, and spintronics [110]. These technologies are inherently probabilistic in nature. Thus, the emerging radical device fabrics provide a natural and fertile ground for the application of stochastic computing.

7.3.5 Theoretical Foundations

Last but not least, theoretical foundations for stochastic design principles need to be established. Such an effort requires a synergy between the seminal works of Shannon on information transfer [111] and Von Neumann on designing reliable systems from unreliable components [43].

REFERENCES

[1] J. M. Rabaey, D. Burke, K. Lutz, and J. Wawrzynek, "Workloads of the future," *IEEE Design & Test of Computers*, vol. 25, no. 4, pp. 358–365, 2008.

[2] M. A. Hanson, H. C. Powell Jr., A. T. Barth, K. Ringgenberg, B. H. Calhoun, J. H. Aylor, and J. Lach, "Body area sensor networks: Challenges and opportunities," *IEEE Computer*, vol. 42, no. 1, pp. 58–65, Jan. 2009.

[3] L. Sun, "The future according to freescale," in *Freescale Technology Forum*, June 2008.

[4] V. Raghunathan, C. Schurgers, S. P. S. Park, and M. B. Srivastava, "Energy-aware wireless microsensor networks," *IEEE Signal Processing Magazine*, vol. 19, no. 2, pp. 40–50, 2002.

[5] T. Karnik, S. Borkar, and V. De, "Sub-90 nm technologies: Challenges and opportunities for CAD," in *Proceedings of the 2002 IEEE/ACM International Conference on Computer-Aided Design*, ser. ICCAD '02, 2002, pp. 203–206.

[6] "Intl. technology roadmap for semiconductors," ITRS, Tech. Rep., 2008.

[7] S. Hanson, B. Zhai, D. Blaauw, D. Sylvester, A. Bryant, and X. Wang, "Energy optimality and variability in subthreshold design," in *Proceedings of the 2006 International Symposium on Low Power Electronics and Design ISLPED 06*, 2006, pp. 363–365.

[8] N. Sturcken, M. Petracca, S. Warren, L. P. Carloni, A. V. Peterchev, and K. L. Shepard, "An integrated four-phase buck converter delivering 1A/mm^2 with 700-ps controller delay and network-on-chip load in 45-nm SOI," in *IEEE Custom Integrated Circuits Conference*, Septemeber 2011, pp. 1–4.

[9] T. Šimunić, L. Benini, and G. D. Micheli, "Energy-efficient design of battery-powered embedded systems," *IEEE Trans. Very Large Scale Integr. Syst.*, vol. 9, no. 1, pp. 15–28, February 2001.

[10] M. Hempstead, N. Tripathi, P. Mauro, G.-Y. Wei, and D. Brooks, "An ultra low power system architecture for sensor network applications," in *ISCA*, 2005, pp. 208–219.

[11] S. Smith, "Power optimization in the connected world," in *Proc. of Int. Conf. on Energy Aware Computing*, Dec. 2010.

[12] A. P. Chandrakasan, S. Sheng, and R. W. Brodersen, "Low-power CMOS digital design," *IEEE Journal of Solid-State Circuits*, vol. 27, no. 4, pp. 473–484, 1992.

[13] S. C. Prasad and K. Roy, "Circuit optimization for minimisation of power consumption under delay constraint," in *Proceedings of the 8th International Conference on VLSI Design*, ser. VLSID '95, 1995, pp. 305–309.

[14] J. Rabaey, "Reconfigurable processing: The solution to low-power programmable DSP," in *Proceedings of the 1997 IEEE International Conference on Acoustics, Speech, and Signal Processing*, ser. ICASSP '97, Washington, DC, USA, 1997, pp. 275–278.

[15] M. Pedram, "Power minimization in IC design: principles and applications," *ACM Trans. Design Autom. Electron. Syst.*, vol. 1, no. 1, pp. 3–56, Jan. 1996.

[16] L. Wei, Z. Chen, K. Roy, Y. Ye, and V. De, "Mixed-V_{th} (MVT) CMOS circuit design methodology for low power applications," in *Proceedings of the 36th annual ACM/IEEE Design Automation Conference*, ser. DAC '99, 1999, pp. 430–435.

[17] M. Borah, R. M. Owens, and M. J. Irwin, "Transistor sizing for low power CMOS circuits," *IEEE Trans. on CAD of Integrated Circuits and Systems*, vol. 15, no. 6, pp. 665–671, 1996.

[18] Q. Wu, M. Pedram, and X. Wu, "Clock-gating and its application to low power design of sequential circuits," *IEEE Trans. on Circuits and Systems I: Fundamental Theory and Applications*, vol. 47, no. 3, pp. 415–420, 2000.

[19] M. Goel and N. R. Shanbhag, "Dynamic algorithm transformations (DAT)-a systematic approach to low-power reconfigurable signal processing," *IEEE Trans. VLSI Syst.*, vol. 7, no. 4, pp. 463–476, 1999.

[20] T. Pering, T. Burd, and R. Brodersen, "The simulation and evaluation of dynamic voltage scaling algorithms," in *Proceedings of the 1998 International Symposium on Low Power Electronics and Design*, ser. ISLPED '98, 1998, pp. 76–81.

[21] J. Zhuo, C. Chakrabarti, and N. Chang, "Energy management of DVS-DPM enabled embedded systems powered by fuel cell-battery hybrid source," in *Proceedings of the 2007 International Symposium on Low Power Electronics and Design*, ser. ISLPED '07, 2007, pp. 322–327.

[22] Y.-H. Lee, Y.-Y. Yang, K.-H. Chen, Y.-H. Lin, S.-J. Wang, K.-L. Zheng, P.-F. Chen, C.-Y. Hsieh, Y.-Z. Ke, Y.-K. Chen, and C.-C. Huang, "A DVS embedded power management for high efficiency integrated SoC in UWB system," *J. Solid-State Circuits*, vol. 45, no. 11, pp. 2227–2238, 2010.

[23] B. H. Calhoun, A. Wang, and A. Chandrakasan, "Modeling and sizing for minimum energy operation in subthreshold circuits," *IEEE Journal of Solid-State Circuits*, vol. 40, no. 9, pp. 1778–1786, 2005.

[24] A. Raychowdhury, B. C. Paul, S. Bhunia, and K. Roy, "Computing with subthreshold leakage: device/circuit/architecture co-design for ultralow-power subthreshold operation," *IEEE Trans. Very Large Scale Integr. Syst.*, vol. 13, no. 11, pp. 1213–1224, Nov. 2005.

[25] B. Zhai, D. Blaauw, D. Sylvester, and K. Flautner, "The limit of dynamic voltage scaling and insomniac dynamic voltage scaling," *IEEE Trans. Very Large Scale Integr. Syst.*, vol. 13, no. 11, pp. 1239–1252, Nov. 2005.

[26] J. Kwong and A. P. Chandrakasan, "Variation-driven device sizing for minimum energy sub-threshold circuits," in *Proceedings of the 2006 International Symposium on Low Power Electronics and Design ISLPED 06*, 2006, pp. 8–13.

[27] A. Wang and A. Chandrakasan, "A 180-mV subthreshold FFT processor using a minimum energy design methodology," *IEEE Journal of Solid-State Circuits*, vol. 40, no. 1, pp. 310–319, 2005.

[28] D. Jeon, M. Seok, C. Chakrabarti, D. Blaauw, and D. Sylvester, "A super-pipelined energy efficient subthreshold 240 ms/s FFT core in 65-nm CMOS," *J. Solid-State Circuits*, vol. 47, no. 1, pp. 23–34, 2012.

[29] C. H.-I. Kim, H. Soeleman, and K. Roy, "Ultra-low-power DLMS adaptive filter for hearing aid applications," *IEEE Trans. on Very Large Scale Integration Systems*, vol. 11, no. 6, pp. 352–357, 2003.

[30] B. H. Calhoun and A. P. Chandrakasan, "Ultra-dynamic voltage scaling (UDVS) using sub-threshold operation and local voltage dithering," *IEEE Journal of Solid-State Circuits*, vol. 41, no. 1, pp. 238–245, 2006.

[31] S. Hanson, M. Seok, Y.-S. Lin, Z. Y. Foo, D. Kim, Y. L. Lee, N. Liu, D. Sylvester, and D. Blaauw, "A low-voltage processor for sensing applications with picowatt standby mode," *IEEE Journal of Solid-State Circuits*, vol. 44, no. 4, pp. 1145–1155, 2009.

[32] B. Zhai, S. Pant, L. Nazhandali, S. Hanson, J. Olson, A. Reeves, M. Minuth, R. Helfand, T. Austin, D. Sylvester, and D. Blaauw, "Energy-efficient subthreshold processor design," *IEEE Trans. Very Large Scale Integr. Syst.*, vol. 17, no. 8, pp. 1127–1137, aug 2009.

[33] S. C. Jocke, J. F. Bolus, S. N. Wooters, T. N. Blalock, and B. H. Calhoun, "A 2.6 μW sub-threshold mixed-signal ECG SoC," in *Proceedings of the 14th ACM/IEEE International Symposium on Low Power Electronics and Design ISLPED 09*, 2009, pp. 117–118.

[34] N. Verma, A. Shoeb, J. Bohorquez, J. Dawson, J. Guttag, and A. P. Chandrakasan, "A micro-power EEG acquisition SoC with integrated feature extraction processor for a chronic seizure detection system," *IEEE Journal of Solid-State Circuits*, vol. 45, no. 4, pp. 804–816, 2010.

[35] J. Kwong and A. P. Chandrakasan, "An energy-efficient biomedical signal processing platform," *J. Solid-State Circuits*, vol. 46, no. 7, pp. 1742–1753, 2011.

[36] R. F. Yazicioglu, S. Kim, T. Torfs, H. Kim, and C. Van Hoof, "A 30 μw analog signal processor ASIC for portable biopotential signal monitoring," *IEEE Journal of Solid-State Circuits*, vol. 46, no. 1, pp. 209–223, 2011.

[37] M. Ashouei, J. Hulzink, M. Konijnenburg, J. Zhou, F. Duarte, A. Breeschoten, J. Huisken, J. Stuyt, H. de Groot, F. Barat, J. David, and J. V. Ginderdeuren, "A voltage-scalable biomedical signal processor running ECG using 13 pJ/cycle at 1 MHz and 0.4 V," in *IEEE International Solid-State Circuits Conference*, February.

[38] S. R. Sridhara et al., "Microwatt embedded processor platform for medical system-on-chip applications," *IEEE Journal of Solid-State Circuits*, vol. 46, no. 4, pp. 721–730, 2011.

[39] K. Roy, S. Mukhopadhyay, and H. Mahmoodi-Meimand, "Leakage current mechanisms and leakage reduction techniques in deep-submicrometer CMOS circuits," *Proceedings of the IEEE*, vol. 91, no. 2, pp. 305–327, 2003.

[40] S. Mutoh, T. Douseki, Y. Matsuya, T. Aoki, S. Shigematsu, and J. Yamada, "1-v power supply high-speed digital circuit technology with multithreshold-voltage CMOS," *IEEE Journal of Solid-State Circuits*, vol. 30, no. 8, pp. 847–854, 1995.

[41] K. Shi and D. Howard, "Challenges in sleep transistor design and implementation in low-power designs," in *Proceedings of the 43rd annual Design Automation Conference*, ser. DAC '06, 2006, pp. 113–116.

[42] T. Chen and S. Naffziger, "Comparison of adaptive body bias (ABB) and adaptive supply voltage (ASV) for improving delay and leakage under the presence of process variation," *IEEE Trans. VLSI Syst.*, vol. 11, no. 5, pp. 888–899, 2003.

[43] J. von Neumann, *Probabilistic logics and the synthesis of reliable organisms from unreliable components.* Princeton, N.J.: Automata Studies, Princeton Univ. Press, 1956.

[44] R. I. Bahar, J. Mundy, and J. Chen, "A probabilistic-based design methodology for nanoscale computation," in *Proceedings of the 2003 IEEE/ACM International Conference on Computer-Aided Design*, ser. ICCAD '03, Washington, DC, USA, 2003, pp. 480–486.

[45] W. Qian, M. D. Riedel, K. Bazargan, and D. J. Lilja, "The synthesis of combinational logic to generate probabilities," in *Proceedings of the 2009 International Conference on Computer-Aided Design*, ser. ICCAD '09, 2009, pp. 367–374.

[46] N. F. Vaidya and D. K. Pradhan, "Fault-tolerant design strategies for high reliability and safety," *IEEE Trans. Comput.*, vol. 42, no. 10, pp. 1195–1206, Oct. 1993.

[47] Y. Tamir and M. Tremblay, "High-performance fault-tolerant VLSI systems using micro rollback," *IEEE Trans. Comput.*, vol. 39, no. 4, pp. 548–554, Apr. 1990.

[48] S. J. Piestrak, "Design of fast self-testing checkers for a class of Berger codes," *IEEE Trans. Comput.*, vol. 36, no. 5, pp. 629–634, May 1987.

[49] C. Winstead and S. Howard, "A probabilistic LDPC-coded fault compensation technique for reliable nanoscale computing," *IEEE Trans. Cir. Sys.*, vol. 56, no. 6, pp. 484–488, June 2009.

[50] N. Jayakumar and S. P. Khatri, "A variation tolerant subthreshold design approach," in *Proceedings of the 42nd annual Design Automation Conference*, ser. DAC '05, 2005, pp. 716–719.

[51] N. Verma, J. Kwong, and A. P. Chandrakasan, "Nanometer MOSFET variation in minimum energy subthreshold circuits," *IEEE Transactions on Electron Devices*, vol. 55, no. 1, pp. 163–174, 2008.

[52] B. Zhai, S. Hanson, D. Blaauw, and D. Sylvester, "Analysis and mitigation of variability in subthreshold design," in *Proceedings of the 2005 International Symposium on Low Power Electronics and Design*, ser. ISLPED '05, 2005, pp. 20–25.

[53] S. Das, D. Roberts, S. Lee, S. Pant, D. Blaauw, T. Austin, K. Flautner, and T. Mudge, "A self-tuning DVS processor using delay-error detection and correction," *IEEE Journal of Solid-State Circuits*, vol. 41, no. 4, pp. 792–804, 2006.

[54] S. Das, C. Tokunaga, S. Pant, W.-H. Ma, S. Kalaiselvan, K. Lai, D. M. Bull, and D. T. Blaauw, "RAZOR-II: In situ error detection and correction for PVT and SER tolerance," *IEEE Journal of Solid-State Circuits*, vol. 44, no. 1, pp. 32–48, 2009.

[55] K. A. Bowman, J. W. Tschanz, N.-S. Kim, J. C. Lee, C. B. Wilkerson, S.-L. L. Lu, T. Karnik, and V. K. De, "Energy-efficient and metastability-immune resilient circuits for dynamic variation tolerance," *IEEE Journal of Solid-State Circuits*, vol. 44, no. 1, pp. 49–63, 2009.

[56] P. Hazucha, T. Karnik, B. A. Bloechel, C. Parsons, D. Finan, and S. Borkar, "Area-efficient linear regulator with ultra-fast load regulation," *IEEE Journal of Solid-State Circuits*, vol. 40, no. 4, pp. 933–940, 2005.

[57] G. Patounakis, Y. W. Li, and K. L. Shepard, "A fully integrated on-chip DCDC conversion and power management system," *IEEE Journal of Solid-State Circuits*, vol. 39, no. 3, pp. 443–451, 2004.

[58] G.-Y. Wei and M. Horowitz, "A fully digital, energy-efficient, adaptive power-supply regulator," *IEEE Journal of Solid-State Circuits*, vol. 34, no. 4, pp. 520–528, 1999.

[59] F. Ichiba, K. Suzuki, S. Mita, T. Kuroda, and T. Furuyama, "Variable supply-voltage scheme with 95%-efficiency dc-dc converter for mpeg-4 codec," in *Proceedings of the 1999 International Symposium on Low Power Electronics and Design*, ser. ISLPED '99, 1999, pp. 54–59.

[60] Y. Ramadass and A. Chandrakasan, "Voltage scalable switched capacitor DC-DC converter for ultra-low-power on-chip applications," in *IEEE Power Electronics Specialists Conference, PESC 2007*, june 2007, pp. 2353–2359.

[61] Y. K. Ramadass and A. P. Chandrakasan, "Minimum energy tracking loop with embedded DCDC converter enabling ultra-low-voltage operation down to 250 mV in 65-nm CMOS," *IEEE Journal of Solid-State Circuits*, vol. 43, no. 1, pp. 256–265, 2008.

[62] Y. Pu, X. Zhang, J. Huang, A. Muramatsu, M. Nomura, K. Hirairi, H. Takata, T. Sakurabayashi, S. Miyano, M. Takamiya, and T. Sakurai, "Misleading energy and performance claims in sub/near threshold digital systems," in *Proceedings of the International Conference on Computer-Aided Design*, ser. ICCAD '10, 2010, pp. 625–631.

[63] Y. Choi, N. Chang, and T. Kim, "DC-DC converter-aware power management for low-power embedded systems," *IEEE Trans. on CAD of Integrated Circuits and Systems*, vol. 26, no. 8, pp. 1367–1381, 2007.

[64] B. Amelifard and M. Pedram, "Design of an efficient power delivery network in an SoC to enable dynamic power management," in *Proceedings of the 2007 International Symposium on Low Power Electronics and Design*, ser. ISLPED '07, 2007, pp. 328–333.

[65] J. Park, D. Shin, N. Chang, and M. Pedram, "Accurate modeling and calculation of delay and energy overheads of dynamic voltage scaling in modern high-performance microprocessors," in *Proceedings of the 16th ACM/IEEE International Symposium on Low Power Electronics and Design*, ser. ISLPED '10, 2010, pp. 419–424.

[66] N. R. Shanbhag, R. A. Abdallah, R. Kumar, and D. L. Jones, "Stochastic computation," in *Proceedings of the 47th Design Automation Conference*, ser. DAC '10, New York, NY, USA, 2010, pp. 859–864.

[67] F. J. Kurdahi, A. M. Eltawil, Y.-H. Park, R. N. Kanj, and S. R. Nassif, "System-level sram yield enhancement," in *Proceedings of the 7th International Symposium on Quality Electronic Design*, ser. ISQED '06, 2006, pp. 179–184.

[68] G. Karakonstantis, G. Panagopoulos, and K. Roy, "HERQULES: system level cross-layer design exploration for efficient energy-quality trade-offs," in *Proceedings of the 16th ACM/IEEE International Symposium on Low Power Electronics and Design*, ser. ISLPED '10, 2010, pp. 117–122.

[69] V. Papirla, A. Jain, and C. Chakrabarti, "Low power robust signal processing," in *Proceedings of the 14th ACM/IEEE International Symposium on Low Power Electronics and Design*, ser. ISLPED '09, 2009, pp. 303–306.

[70] R. Hegde and N. R. Shanbhag, "Soft digital signal processing," *IEEE Trans. Very Large Scale Integr. Syst.*, vol. 9, no. 6, pp. 813–823, Dec. 2001.

[71] R. Hegde and N. R. Shanbhag, "A voltage overscaled low-power digital filter IC," *IEEE Journal of Solid-State Circuits*, vol. 39, no. 2, pp. 388–391, 2004.

[72] G. V. Varatkar and N. R. Shanbhag, "Error-resilient motion estimation architecture," *IEEE Transactions on Very Large Scale Integration Systems*, vol. 16, no. 10, pp. 1399–1412, 2008.

[73] R. A. Abdallah and N. R. Shanbhag, "Error-resilient low-power Viterbi decoder architectures," *IEEE Trans Signal Processing*, vol. 57, no. 12, pp. 4906–4917, 2009.

[74] G. V. Varatkar, S. Narayanan, N. R. Shanbhag, and D. L. Jones, "Stochastic networked computation," *IEEE Transactions on Very Large Scale Integration Systems*, vol. 18, no. 10, pp. 1421–1432, 2010.

[75] P. J. Huber, *Robust Statistics*. Wiley, 1981, vol. 1, no. 3.

[76] E. P. Kim, D. J. Baker, S. Narayanan, D. L. Jones, and N. R. Shanbhag, "Low power and error resilient PN code acquisition filter via statistical error compensation," in *IEEE Custom Integrated Circ. Conf. (CICC)*, 2011, pp. 1–4.

[77] S. Mitra, N. R. Saxena, and E. J. McCluskey, "A design diversity metric and analysis of redundant systems," *IEEE Trans. Comput.*, vol. 51, no. 5, pp. 498–510, May 2002.

[78] E. P. Kim and N. R. Shanbhag, "Soft N-modular redundancy," *IEEE Trans. Comput.*, vol. 61, no. 3, pp. 323–336, Mar. 2012.

[79] B. Shim, S. R. Sridhara, and N. R. Shanbhag, "Reliable low-power digital signal processing via reduced precision redundancy," *IEEE Transactions on Very Large Scale Integration Systems*, vol. 12, no. 5, pp. 497–510, 2004.

[80] H. Mahmoodi, S. Mukhopadhyay, and K. Roy, "Estimation of delay variations due to random-dopant fluctuations in nanoscale CMOS circuits," *IEEE Journal of Solid-State Circuits*, vol. 40, no. 9, pp. 1787–1796, 2005.

[81] J. Fayn and P. Rubel, "Toward a personal health society in cardiology," *IEEE Trans. Info. Tech. Biomed.*, vol. 14, no. 2, pp. 401–409, Mar. 2010.

[82] N. M. Arzeno, Z.-D. Deng, and C.-S. Poon, "Analysis of first-derivative based QRS detection algorithms," *IEEE Transactions on Biomedical Engineering*, vol. 55, no. 2, pp. 478–484, 2008.

[83] M. G. Tsipouras, D. I. Fotiadis, and D. Sideris, "An arrhythmia classification system based on the RR-interval signal," *Artificial Intelligence in Medicine*, vol. 33, no. 3, pp. 237–250, 2005.

[84] "The global burden of disease," World Health Organization, Geneva, Tech. Rep., 2008.

[85] V. Fuster and B. Kelly, *Promoting cardiovascular health in the developing world*. Washington DC: The National Academies Press, 2010.

[86] J. Pan and W. J. Tompkins, "A real-time QRS detection algorithm," *IEEE Transactions on Biomedical Engineering*, vol. 32, no. 3, pp. 230–236, 1985.

[87] A. Amann, R. Tratnig, and K. Unterkofler, "Reliability of old and new ventricular fibrillation detection algorithms for automated external defibrillators," *BioMedical Engineering Online*, vol. 4, no. 1, p. 60, 2005.

[88] PhysioNet, "MIT-BIH arrhythmia database," [Online]. Available: http://www.physionet.org/physiobank/database/mitdb.

[89] B. U. Kohler, C. Hennig, and R. Orglmeister, "The principles of software QRS detection," *IEEE Engineering in Medicine and Biology Magazine*, vol. 21, no. 1, pp. 42–57, 2002.

[90] K. V. Surez, J. C. Silva, Y. Berthoumieu, P. Gomis, and M. Najim, "ECG beat detection using a geometrical matching approach," *IEEE Transactions on Biomedical Engineering*, vol. 54, no. 4, pp. 641–650, 2007.

[91] G. M. Friesen, T. C. Jannett, M. A. Jadallah, S. L. Yates, S. R. Quint, and H. T. Nagle, "A comparison of the noise sensitivity of nine QRS detection algorithms," *IEEE Transactions on Biomedical Engineering*, vol. 37, no. 1, pp. 85–98, 1990.

[92] "References testing and reporting performance results of cardiac rhythm and ST segment measurement algorithms," AAMI Recommended Practice/American National Standard, Tech. Rep., 1998.

[93] H. Arbetter, R. Erickson, and D. Maksimovid, "DC-DC converter design for battery-operated systems," in *IEEE Power Electronics Specialists Conference*, 1995, pp. 103–109.

[94] V. Kursun, S. G. Narendra, V. K. De, and E. G. Friedman, "Low-voltage-swing monolithic DCDC conversion," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 51, no. 5, pp. 241–248, 2004.

[95] R. A. Abdallah, Y.-h. Lee, and N. R. Shanbhag, "Timing error statistics for energy-efficient robust DSP systems," in *Proceedings of the conference on Design, automation and test in Europe*, ser. DATE, 2011, pp. 1–4.

[96] J. Erfanian, S. Pasupathy, and G. Gulak, "Reduced complexity symbol detectors with parallel structure for ISI channels," *IEEE Transactions on Communications*, vol. 42, no. 234, pp. 1661–1671, Feb./Mar./Apr. 1994.

[97] W.-H. Chen, C. H. Smith, and S. C. Fralick, "A fast computational algorithm for the discrete cosine transform," *IEEE Transactions on Communications*, vol. 25, no. 9, pp. 1004–1009, 1977.

[98] E. P. Kim and N. R. Shanbhag, "Soft NMR: Analysis and application to DSP systems," in *Proceedings of the 2010 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 10)*, 2010, pp. 1494–1497.

[99] S. Kullback and R. A. Leibler, "On information and sufficiency," *Ann. Mathematical Statistics*, vol. 22, no. 1, Mar. 1951.

[100] Y. Liu, T. Zhang, and K. K. Parhi, "Computation error analysis in digital signal processing systems with overscaled supply voltage," *IEEE Trans. Very Large Scale Integr. Syst.*, vol. 18, no. 4, pp. 517–526, Apr. 2010.

[101] J. H. Lala and R. E. Harper, "Architectural principles for safety-critical real-time applications," *Proceedings of the IEEE*, vol. 82, no. 1, pp. 25–40, 1994.

[102] Y. Tamir and C. H. Sequin, "Reducing common mode failures in duplicate module," in *Proc. IEEE Int. Conf. Computer Design*, 1984, pp. 302–307.

[103] G. Taylor, "Energy-efficient circuit design and the future of power delivery," in *IEEE Electrical Performance of Electronic Packaging and Systems Conference, EPEPS'09*, October 2009.

[104] S. Rusu, S. Tam, H. Muljono, J. Stinson, D. Ayers, J. Chang, R. Varada, M. Ratta, S. Kottapalli, and S. Vora, "A 45 nm 8-core enterprise Xeon processor," *J. Solid-State Circuits*, vol. 45, no. 1, pp. 7–14, 2010.

[105] R. Kumar and G. Hinton, "A family of 45-nm IA processors," in *IEEE International Solid-State Circuits Conference, ISSCC'09*, February 2009, pp. 58–59.

[106] M. Bohr, "The new era of scaling in an SoC world," in *IEEE International Solid-State Circuits Conference, ISSCC'09*, February 2009, pp. 23–28.

[107] S. Vangal, J. Howard, G. Ruhl, S. Dighe, H. Wilson, J. Tschanz, D. Finan, P. Iyer, A. Singh, T. Jacob, and et al., "An 80-tile 1.28 TFLOPS network-on-chip in 65-nm CMOS," in *IEEE International Solid-State Circuits Conference, ISSCC'07*, February 2007, pp. 98–99.

[108] V. J. Reddi, S. Kanev, W. Kim, S. Campanoni, M. D. Smith, G.-Y. Wei, and D. Brooks, "Voltage smoothing: Characterizing and mitigating voltage noise in production processors via software-guided thread scheduling," in *Proceedings of the 2010 43rd Annual IEEE/ACM International Symposium on Microarchitecture*, ser. MICRO '43. Washington, DC, USA: IEEE Computer Society, 2010, pp. 77–88.

[109] V. Guruswami and P. Indyk., "Expander-based constructions of efciently decodable codes," in *Proceedings of the IEEE Symposium on Foundations of Computer Science*, 2001, pp. 658–667.

[110] H.-S. P. Wong, "Beyond the conventional transistor," *IBM J. Res. Dev.*, vol. 46, no. 2-3, pp. 133–168, Mar. 2002.

[111] C. E. Shannon, "A mathematical theory of communication," *Bell System Technical Journal*, vol. 27, pp. 379–423 and 623–656, July and October 1948.